

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

DIPLOMOVÁ PRÁCE



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

CNC ŘÍZENÍ FRÉZKY

MILLING MACHINE CNC CONTROL

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Jan Krejčí

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. Zdeněk Bradáč, Ph.D.

BRNO 2021

Diplomová práce

magisterský navazující studijní program **Kybernetika, automatizace a měření**

Ústav automatizace a měřicí techniky

Student: Bc. Jan Krejčí

ID: 195368

Ročník: 2

Akademický rok: 2020/21

NÁZEV TÉMATU:

CNC řízení frézky

POKYNY PRO VYPRACOVÁNÍ:

Zrealizujte digitální řízení 3-osé frézky se zpětnovazebním řízením os. Navrhněte elektro-výbavu frézky včetně řídicího systému. Vybavte nezbytnou sensorikou, ovládacími a bezpečnostními prvky. Vytvořte vhodné programové vybavení a propojení do informačního systému továrny.

1. Proveďte literární a internetovou rešerši.
2. Navrhněte koncept digitalizace manuální 3osé frézky.
3. Navrhněte a zrealizujte elektroniku řízení. Vybavte zpětnovazebním řízením os. Oživte systém a vyřešte nastavení HOME pozice.
4. Implementujte programové vybavení řídicího systému frézky a její napojení do výrobního systému.
5. Demonstrujte funkčnost, programování obrábění a vyhodnoťte dosažené výsledky.

DOPORUČENÁ LITERATURA:

Pavel Herout: Učebnice jazyka C, KOPP, 2004, IV. přepracované vydání, ISBN 80-7232-220-6

Patrick Hood-Daniel. Build Your Own CNC Machine. Apress, 2009. ISBN: 978-1430224891.

Dle pokynů vedoucího práce.

Termín zadání: 8.2.2021

Termín odevzdání: 17.5.2021

Vedoucí práce: doc. Ing. Zdeněk Bradáč, Ph.D.

doc. Ing. Petr Fiedler, Ph.D.
předseda rady studijního programu

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

Abstrakt

Tato diplomová práce se zabývá přestavěním manuální vrtačkofrézky BF20L na CNC frézku, její 3D model a případné PLC řízení. Všechny osy jsou poháněny krokovými motory, s enkodéry, připojenými do řídících karet a průmyslového počítače od společnosti Beckhoff. Pro toto řízení je vytvořen kompletní rozvaděč, doplněný o nakreslené elektrotechnické schéma zapojení. Programové řízení společně s vizualizací je realizováno v softwaru TwinCAT 3. Frézku lze řídit manuálně i automaticky pomocí programu.

Klíčová slova

Vrtačkofrézka, CNC řízení, krokový motor, rozvaděč, elektrotechnické schéma zapojení, 3D model, Beckhoff, PLC

Abstract

This diploma thesis deals with rebuild of manual drill-mill machine BF20L into CNC mill machine, 3D model of that and eventual control using PLC. All axes are powered by stepper motors, with encoders, connected to control cards and industrial computer from company Beckhoff. For this control is made full switchboard, supplemented by electrical wiring diagram. Program control along with visualization is made in software TwinCAT 3. Drill-mill machine can be controlled manually or automatically by program.

Keywords

Drill-mill machine, CNC control, stepper motor, switchboard, electrical wiring diagram, 3D model, Beckhoff, PLC

Citace tištěné práce:

KREJČÍ, Jan. *CNC řízení frézky*. Brno, 2021. Dostupné také z: <https://www.vutbr.cz/studenti/zav-prace/detail/134826>. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky. Vedoucí práce Zdeněk Bradáč.

Citace elektronického zdroje:

KREJČÍ, Jan. *CNC řízení frézky* [online]. Brno, 2021 [cit. 2021-04-08]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/134826>. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky. Vedoucí práce Zdeněk Bradáč.

Prohlášení autora o původnosti díla

Jméno a příjmení studenta:	<i>Jan Krejčí</i>
VUT ID studenta:	<i>195368</i>
Typ práce:	<i>Diplomová práce</i>
Akademický rok:	<i>2020/21</i>
Téma závěrečné práce:	<i>CNC řízení frézky</i>

Prohlašuji, že svou závěrečnou práci jsem vypracoval samostatně pod vedením vedoucího závěrečné práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne: **14. května 2021**

.....

Jan Krejčí, podpis

Poděkování

Tímto bych chtěl poděkovat vedoucímu diplomové práce panu doc. Ing. Zdeňku Bradáčovi, Ph.D. za odborné vedení, trpělivost, ochotu, za pomoc při řešení problémů a vždy veselého ducha. Díky patří také Ing. Ondřeji Baštánovi za praktické vedení a pomoc. V neposlední řadě bych chtěl poděkovat všem kolegům ze společnosti Beckhoff Česká Republika s.r.o., mé přítelkyni a rodině.

V Brně dne: **14. května 2021**

.....

Jan Krejčí, podpis

OBSAH

Úvod	12
1 Úvod do Problematiky CNC strojů	13
1.1 Vývoj CNC strojů	14
1.2 Souřadnicový systém stroje	15
1.3 Vztažné body CNC stroje	16
1.4 CNC program	17
1.4.1 Stavba CNC programu	18
1.4.2 Důležité G a M funkce	19
1.4.3 Programy pro generaci kódu	20
1.5 Popis funkce řídicích systémů CNC	21
1.5.1 Ovládací režimy	22
1.5.2 Současně dostupné řídicí systémy	22
2 Analýza výchozího stavu frézky Optimum Vario BF 20L	24
2.1 Popis ovládání	25
2.2 Technické parametry	26
3 Hardware	27
3.1 Napájení	27
3.1.1 Použité napájecí prvky	27
3.2 Elektrické pohony	29
3.2.1 Servomotor	29
3.2.2 Krokový motor	30
3.2.3 Použité pohony	32
3.3 Řídicí systém	32
3.3.1 Programovatelný logický automat	32
3.3.2 Použité řízení	33
3.4 Ostatní použité elektrické zařízení	35
3.4.1 MPG ovladač	35
3.4.2 Čidlo korekce délky nástroje	35
3.4.3 Koncové čidla	36
3.4.4 Dotyková obrazovka	36

4	Realizace konstrukce a el. zapojení	37
4.1	Mechanická rekonstrukce	37
4.1.1	Přípevnění motorů a enkodérů	38
4.1.2	Celkový vzhled a upevnění frézky	40
4.2	3D model frézky	40
4.2.1	Držáky koncových snímačů	41
4.2.2	Vytvořený model krytu motoru	41
4.2.3	Přepojení motorů	42
4.3	Elektrické zapojení	43
4.3.1	Hlavní rozvaděč	44
4.3.2	Připojení IPC a coupleru	45
4.3.3	Připojení vstupů	45
4.3.4	Připojení výstupů	46
4.3.5	Připojení motoru ke kartě EL7041	47
4.3.6	Připojení ručního ovladače	48
4.3.7	Funkce a popis vnějšku hlavního rozvaděče	49
4.3.8	Vedlejší rozvaděč.....	50
4.3.9	Náhled fyzického zapojení řízení.....	50
5	TwinCAT 3, a jeho nastavení	51
5.1	TwinCAT 3.....	51
5.1.1	Vytvoření projektu	53
5.2	Struktura TwinCAT projektu.....	53
5.2.1	Využití licence a knihovny	54
5.3	Deklarace a linkování I/O proměnných.....	55
5.4	Nastavení parametrů pohonů	56
5.4.1	Nastavení parametrů terminálů.....	57
5.4.2	Nastavení parametrů os.....	58
5.5	Boot projekt	60
5.6	Nastavení HMI serveru.....	60
5.7	EtherCAT a výběr sběrnic	62
6	Softwarové zpracování	63
6.1	Požadavky na řízení s návrhem HMI	63

6.2	Struktura programového řízení.....	65
6.2.1	Výměna proměnných s HMI.....	66
6.3	Základní funkční bloky pohybů	67
6.3.1	Funkční blok motorů	67
6.3.2	Funkční blok pohybů	69
6.4	Manuální řízení.....	70
6.4.1	Manuální řízení z vizualizace.....	70
6.4.2	Manuálním řízení ručním ovladačem	70
6.5	Automatické řízení	71
6.5.1	Stavový automat	72
6.5.2	JavaScript funkce.....	74
6.6	Parametry a dodatečné ovládání	76
6.6.1	Nástrojová korekce	77
6.7	Programování informačního kanálu.....	78
6.7.1	Vykreslování pohybu.....	78
6.7.2	Informace o motorech a terminálu.....	79
6.7.3	Eventlogger	80
6.8	Ostatní vytvořené funkční vybavení	81
6.8.1	Vypnutí a restart IPC	81
6.8.2	Vybavení vizualizace.....	82
7	Popis ovládání	83
7.1	Popis pohotovostního ovládání	84
7.2	Popis vizualizace – manuální ovládání	85
7.3	Popis vizualizace – automatické ovládání.....	85
7.4	Popis ovládání doprovodných prvků.....	86
8	Uvedení do provozu	89
	Závěr	91
	Seznam literatury	92
	Seznam příloh.....	95

SEZNAM OBRÁZKŮ

Obr. 1.1: Blokové schéma obráběcího CNC stroje [4].....	13
Obr. 1.2: Definování kartézské soustavy souřadnic [4]	15
Obr. 1.3: Vztažné body CNC stroje [2]	16
Obr. 1.4: Stavba jednoho bloku G-kódu složeného z příkazů [2].....	18
Obr. 1.5: Příklad vývojového prostředí Solidworks.....	21
Obr. 1.6: Sinumerik 840 [8]	22
Obr. 1.7: Rozhraní softwaru Mach4 [9]	23
Obr. 1.8: CNC ovládací set [10]	23
Obr. 2.1: Optimum Vario BF 20L [12].....	24
Obr. 2.2: Ovládací panel vřetene	25
Obr. 3.1: Hlavní vypínač [13].....	27
Obr. 3.2: Hlavní zdroj MW [14]	28
Obr. 3.3: Spínací zdroj CarSPA	28
Obr. 3.4: Pojistkové pouzdro [15]	28
Obr. 3.5: Servomotor [16]	29
Obr. 3.6: Krokový motor [16]	30
Obr. 3.7: Varianty zapojení krokových motorů [21]	31
Obr. 3.8: Použitý krokový motor [22]	32
Obr. 3.9: IPC C6015-0010 [10].....	33
Obr. 3.10: Použité I/O karty	34
Obr. 3.11: Ruční ovladač ISM1468 [27].....	35
Obr. 3.123: Čidlo korekce délky nástroje TS32 [28].....	35
Obr. 3.13: Referenční čidla[29]	36
Obr. 3.14: Dotyková obrazovka[30]	36

Obr. 4.1: Retrofit osy X.....	37
Obr. 4.2: Montáž kitu osy Y	38
Obr. 4.5: Celkový vzhled frézky	39
Obr. 4.3: Nástavec motoru.....	39
Obr. 4.4: Fixace motoru a enkodéru	39
Obr. 4.6: 3D model retrofitu frézky	40
Obr. 4.7: Modely držáků snímačů X, Y, Z (zleva)	41
Obr. 4.8: Model motoru s krytem	42
Obr. 4.9: DPS přepojení motoru.....	42
Obr.4.10: Osazená DPS na krytu	42
Obr. 4.11: Blokový návrh rozvaděčů (hlavní vlevo, vedlejší vpravo)	43
Obr. 4.12: Zapojení hlavního rozvaděče	44
Obr. 4.13: Připojení IPC a EtherCAT coupleru.....	45
Obr. 4.14: Připojení terminálu krokového motoru EL7041	47
Obr. 4.16: Zapojení ručního ovladače (MPG).....	48
Obr. 4.15: Značení MPG ovladače [27]	48
Obr. 4.17: Napájecí rozvaděč	49
Obr. 4.18: Čelní strana hlavního rozvaděče	49
Obr. 4.19: Elektrotechnické schéma připojení terminálů.....	50
Obr. 5.1: Spouštění TwinCAT XAE	53
Obr. 5.2: Struktura TC projektu.....	53
Obr. 5.3: Proces linkování proměnných.....	55
Obr. 5.4: Struktura os a jejich nastavení.....	56
Obr. 5.5: Hardwarová konfigurace (vlevo), Startup list parametrů (vpravo)	57
Obr. 5.6: Propojení PLC a HMI serveru	61

Obr. 5.7: Publish to TwinCAT HMI Server	61
Obr. 6.1: Návrh vizualizace.....	63
Obr. 6.2: Návrh automatického řízení (vlevo), manuálního (vpravo)	64
Obr. 6.3: Struktura PLC projektu	65
Obr. 6.4: FB_Engine.....	67
Obr. 6.5: Stavy referování [37]	68
Obr. 6.6: Stavový automat automatického řízení	73
Obr. 6.7: Stavový automat určení délkové korekce nástroje	78
Obr. 6.8: Vykreslení pohybu.....	79
Obr. 6.9: HMI zobrazení událostí	81
Obr 6.10: Eventy HMI.....	82
Obr. 7.1: Náhled na celkovou vizualizaci	83
Obr. 7.2: Pohotovostní ovládání	84
Obr. 7.3: Ovládání manuálního řízení	84
Obr. 7.4: Ovládání automatického řízení	85
Obr. 7.5: Ovládání motorů a vřetene	86
Obr. 7.6: Override, funkce nástroje a souřadnicový systém.....	87
Obr. 7.7: Aktuální souřadnice, rychlosti a stavy os	88
Obr. 8.1: Vůle v převodu osa Y [37]	90

SEZNAM TABULEK

Tab. 1.1: Často používané g-funkce [1]	20
Tab. 1.2: Často používané m-funkce [1]	20
Tab. 2.1: Technické parametry frézky [11]	26
Tab. 3.1: Výhody a nevýhody krokových motorů [19]	30
Tab. 4.1: Seznam digitální vstupů	46
Tab. 4.2: Seznam digitálních výstupů	46
Tab. 5.1: Parametry terminálu EL7041	57
Tab. 5.2: Nastavené parametry os	58
Tab. 5.3: Nastavené parametry „Enc“	59
Tab. 6.1: Vyčítané informace o terminálu EL7041	80
Tab. 8.1: Polarity motorů a enkodérů v programu	89

ÚVOD

Tato diplomová práce se zabývá přestavěním manuální vrtačkofrézky BF20L na CNC frézku a jejího řízení pomocí PLC. Součástí této práce je také teoretický rozbor spojený s přestavbou a řízením frézky, navržení vhodných komponent pro řízení a v neposlední řadě návrh a realizace celého řídicího programu.

K přestavbě je využit speciální „kit“ pomocí kterého jsou všechny osy poháněny krokovými motory doplněné o enkodéry. Frézka je doplněna limitními snímači. Pro jejich uchycení jsou vytvořeny modely v programu Solidworks, společně s 3D modelem celé frézky. Pro řízení je použit průmyslový počítač C6015-0010 se softwarovým PLC společně s I/O terminály a terminály pro řízení krokových motorů EL7041 od společnosti Beckhoff. Elektronika řízení je implementována do průmyslových rozvaděčů. Vytvořeno je i elektrotechnické schéma zapojení v programu ProfiCAD.

Programové řízení je vytvořeno v softwaru TwinCAT3 a jeho funkcích. Naprogramováno je pohotovostní řízení tlačítky z rozvaděče, manuální řízení za pomoci vizualizace, jehož součástí je i ruční ovladač, a dále také automatické řízení, které umožní obrábění pomocí předem definovaného programu. Pro tato řízení je vytvořena vizualizace. Samotné původní ovládání frézky je stále použitelné při deaktivaci motorů. Na vytvoření programu je použit standardní PLC jazyk strukturovaný text s lehkou nadstavbou objektového programování. Vytvořená vizualizace řízení je doplněna funkcemi, vytvořenými jazykem JavaScript.

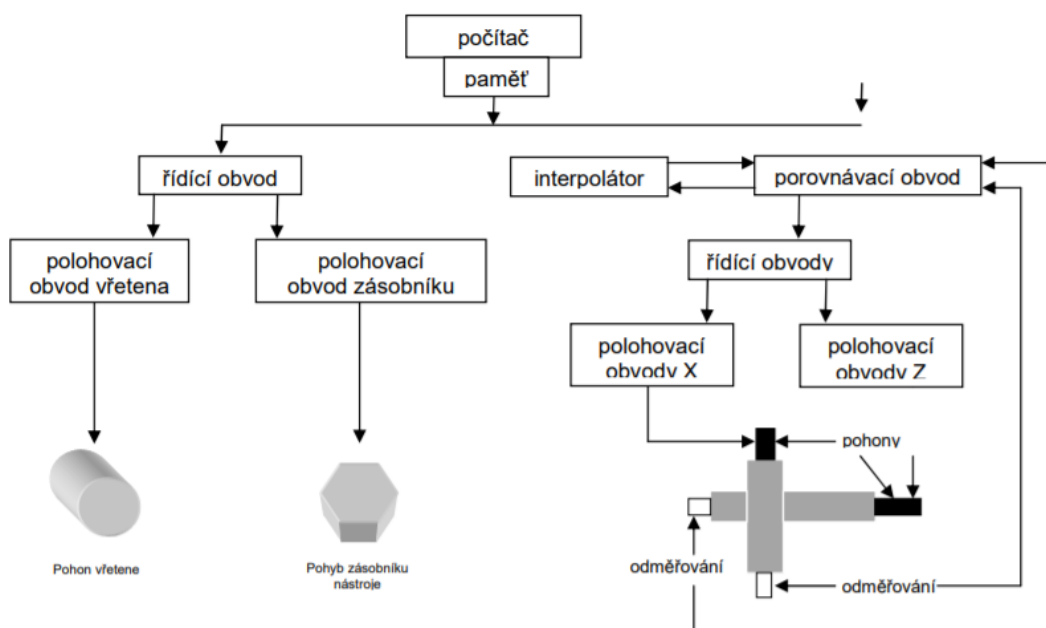
1 ÚVOD DO PROBLEMATIKY CNC STROJŮ

Tématem této části je v případě neznalosti uvést čtenáře do oblasti CNC strojů a jejich programování.

Základní otázka zní: Co je NC nebo CNC stroj?

Jedná se o číslicově řízený stroj (numeric control) či počítačem řízený stroj (computer numeric control). Tyto stroje přispívají k automatizaci procesu, ke zvýšení produktivity práce ruku v ruce se snížením nákladů a přispívají k pružnosti práce. Ovládání je zde zajištěno programem, který je vytvořen programátorem. Toto programování klade v současnosti a v souvislosti s 5D obráběcími centry a věci s nimi spojené, čím dál vyšší požadavky na kvalifikaci a znalosti programátora. Na druhou stranu nevyžaduje téměř žádné znalosti od pracovníka, který bude na stroji upevňovat a odebírat obrobky. [1], [2]

Zjednodušené blokové schéma obráběcího CNC stroje za účelem lepšího pochopení jednotlivých činností je nastíněno na obrázku 1.1.



Obr. 1.1: Blokové schéma obráběcího CNC stroje [4]

- **Počítač** – Většinou se jedná o průmyslový počítač, do něhož je nahrán řídicí systém, popřípadě elektronika vyrobená přímo pro řízení. V interakci s řídicím panelem je možné řídit stroj v různých režimech ovládání,

upravovat nebo dokonce vytvářet požadovaný CNC program.

- **Řídicí obvody** – Zásadou těchto obvodů se přetvářejí logické signály na silnoproudé elektrické signály, kterými jsou přímo ovládány jednotlivé části stroje, jako jsou např.: motory posuvů či vřeteno.
- **Interpolátor** – Je prvek řešící dráhu nástroje zadanou pomocí geometrie programu a délkových a rádiusových korekcí. Počítá tedy pohyb nástroje mezi jednotlivými bloky programu. Interpolace může být lineární či kruhová.
- **Porovnávací obvod** – Tyto obvody zastávají úlohu zpětné vazby, která přenáší informace od jednotlivých os a sledují, v porovnání se zadaným programem, zda se nástroj nachází na požadovaných souřadnicích. Korekce rozdílů je provedena dodatečným posuvem motorů. Nutnou podmínkou pro správnou funkci porovnávacího obvodu je odměřování souřadnic stroje.
- **Řídicí panel** – Dnes často nazýváno HMI, což v překladu znamená „human machine interface“, vytváří rozhraní mezi počítačem a člověkem. Každý panel se liší v závislosti na výrobcu. Vždy se zde však nachází obrazovka, může být i dotyková, a vstupní a výstupní prvky, tzn. ovládací tlačítka a prvky a signalizační prvky. [1], [4]

1.1 Vývoj CNC strojů

Dlouhodobým směrem rozvoje ve všech výrobních odvětvích je automatizace výrobního procesu. Automatizací obráběcího stroje, jako například frézky, dostáváme CNC stroj řízený programem z počítače, který přispívá ke snížení nákladů a zároveň ke zvyšování produktivity práce.

Přelom automatizace se datuje na počátek 50. let 20. století v Americe. Zde byly vyvinuty první numericky řízené stroje neboli NC stroje.

Dalším milníkem byla 70. Léta, která přinesla již NC systémy na bázi základních integrovaných desek namísto elektronkových obvodů.

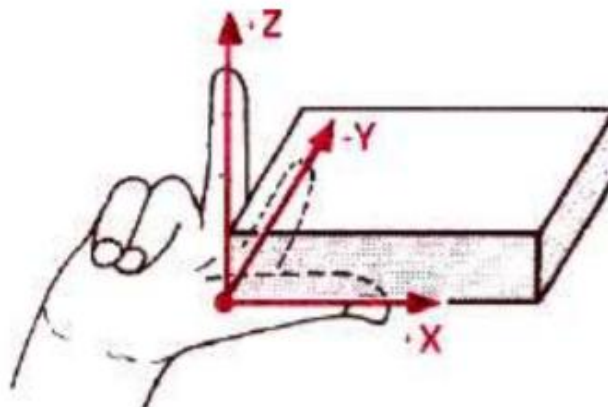
Za největší rozmach je pak považován průběh 80. a 90. let, kde se již tyto CNC stroje velice podobali dnešním. Nacházeli se zde již zásobníky, podavače a senzory. Dnešní CNC jsou pak oproti těmto vyspělejší technologickými možnostmi a parametry, což vede na přesnější a rychlejší výrobu.

Historicky se vývoj číslicově řízených strojů dělí na 4 základní etapy neboli generace:

1. **Vývojová generace NC strojů** - Vyznačují se nejjednodušší stavbou založenou na konstrukci konvenčních strojů doplněné o číslicově řídicí systém. To umožňovalo řízení v pravoúhlých souřadnicích a vždy pouze v jednom směru. Pro dnešní dobu nevyhovující.
2. **Vývojová generace NC strojů** – Stroje jsou již vybaveny pohony, zásobníky nástrojů a dovolují řízení v základních pracovních cyklech. Systémy umožňují souvislé řízení ve dvou osách souvisle, editaci programu a jsou doplněny o paměť.
3. **Vývojová generace NC strojů** – Konstrukce je uzpůsobena na zvýšení produkce, přesnosti, množství nástrojů v zásobníku a k využití senzorů pro zpětnou vazbu od pohonů a mechanismů. Umožňuje souvislé řízení ve třech osách. Výroba součástky od vstupu po výstup je realizována pomocí počítačového řízení.
4. **Vývojová generace NC strojů** – Stroje jsou považovány za obráběcí centra, které umožňují souvislé řízení ve více než třech osách. Uplatňují se zde vědecké poznatky jako kontrola rozměrů obroku, laserové seřízení nástroje a zpracování pomocí systému CAD/CAM. [2], [3]

1.2 Souřadnicový systém stroje

Základním využívaným souřadným systémem ve většině CNC strojů je **kartézský systém souřadnic**. Tento systém je pravoúhlý, pravotočivý s osami **X**, **Y**, **Z**. Jeho znázornění si můžeme představit pomocí pravidla pravé ruky, viz Obr. 1.2.



Obr. 1.2: Definování kartézské soustavy souřadnic [4]

Rotační pohyb kolem těchto os interpretují osy A, B, C. Také platí, že osa pracovního vřetene je rovnoběžná s osou Z, přičemž kladný směr je od obrobku k nástroji. Hodnoty se mohou vyskytovat i v záporném poli souřadnic. Písmena IJK a UVW označují tzv. doplňkové či sdružené osy, ty jsou použity v případě více řízených pohybů v jednom směru. Kartézský systém je nutný pro správné řízení stroje, kde se nástroje pohybuje v závislosti na zadaných příkazech v CNC programu nebo příkazech z panelu CNC stroje.

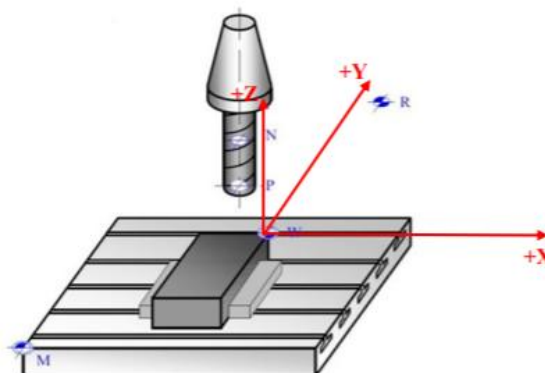
I když je u většiny CNC strojů využíván kartézský systém souřadnic, tak navzdory tomu je často používám i **polární souřadný systém**. Jehož poloha nástroje je určována pomocí: pólu (výchozí bod), rádiu (poloměr) a úhlu natočení.

Základní pravidla pro orientaci os v prostoru u obráběcích strojů:

- vychází se od nehybného obrobku
- vždy musí být definována osa X
- osa X leží v upínací rovině obrobku nebo je s ní rovnoběžná
- osa Z je totožná nebo rovnoběžná s osou pracovního vřetena
- kladný smysl os je od obrobku k nástroji
- pokud jsou na stroji další doplňkové pohyby v osách X,Y,Z, jsou tyto pohyby označeny písmeny UVW a IJK
- pokud se obrobek pohybuje proti nástroji, označují se takové osy X', Y', Z' [1], [2], [4], [7]

1.3 Vztažné body CNC stroje

Po spuštění CNC stroje se aktivuje jeho souřadnicový systém, důležité jsou však i vztažné (nulové) body, obr. 1.3. Tyto body určují vzájemnou polohu soustavy stroj-obrobek-nástroj. Dělí se na body dané výrobcem, jejichž polohu nelze měnit a body, které volí programátor v závislosti na obráběné součásti.



Obr. 1.3: Vztažné body CNC stroje [2]

- **M – nulový bod stroje:** Je počátkem souřadného systému pracovního prostoru stroje. Z pohledu obsluhy bývá u frézky tento bod většinou umístěn vlevo a vepředu, v krajních polohách os X, Y, Z. Tento bod je dán výrobcem a nelze ho měnit.
- **R – referenční bod stroje:** Je to výrobcem zvolené místo na stroji, obvykle je to maximální vzdálenost od nulového bodu stroje v pracovním prostoru stroje, které je nastaveno zreferováním po staru stroje. Při absolutním zadávání souřadnic nemůže stroj bez referenčního bodu pracovat.
- **W – nulový bod obrobku:** Je počátkem souřadného systému obrobku. Poloha je volena obsluhou nebo programátorem a lze měnit i v průběhu programu. U tvarově souměrných obrobků se volí obvykle v ose souměrnosti.
- **P – nulový bod nástroje:** Jedná se o bod ležící na špičce nástroje. Je nutný pro stanovení délkové a poloměrové korekce.[1], [2], [3], [4]

1.4 CNC program

Řídicí program je soubor informací, které podrobně popisují činnost stroje. V minulosti byl program pro stroj uchován na dřevěném štítku, později však s rozvojem technologií se program začal vytvářet číslicově. Dnes je již strojový kód pod záštitou normy ISO, ale mnoho výrobců/programátorů implementuje do kódu i vlastní řešení. K programování je nejčastěji používán kartézský systém souřadnic. Nejběžnějším programováním je pak v těchto souřadnicích absolutní programování.

Možné způsoby programování:

- **Absolutní programování:** programátorem je určen počátek kartézských souřadnic. V blocích programu se nadále popisuje pohyb nástrojem, tzn.: ze kterých a do kterých souřadnic se má v každé ose nástroj posunout.
- **Přírůstkové programování:** neboli inkrementální, počátkem souřadnic je většinou špička nástroje. V bloku programu se popisuje, o kolik se má pohybovat nástroj ze startovací do cílové polohy.
- **Programování v polárních souřadnicích:** z určitého bodu souřadnic se stanovuje délka pohybu U nástroje a úhel A. Praktické využití by mohlo být např. vrtání otvorů po kružnici.
- **Parametrické programování:** definuje parametry v blocích programu, ve kterých jsou vyjádřeny matematické funkce. [1]

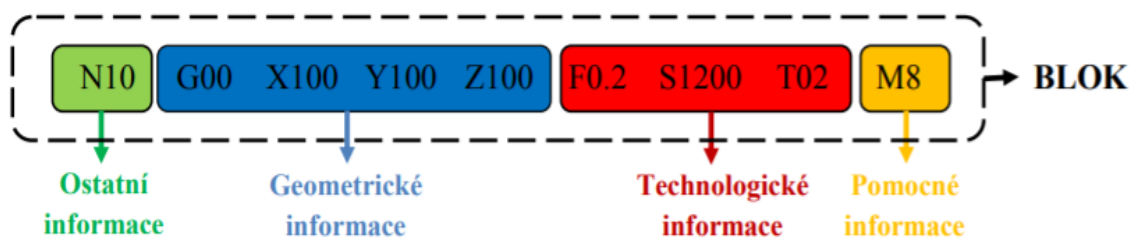
1.4.1 Stavba CNC programu

Programovací kód je daný jednoduchou skladbou slov seřazených do řetězce. Vyobrazeno na obrázku 1.4.

Informace, které program obsahuje, lze rozdělit na následující základní druhy:

- **Geometrické informace:** popisují dráhy nástrojů (způsob pohybu), jako například pracovní posuv, rychloposuv, posuv po přímce nebo kruhovém oblouku. Jsou dány tvarem (rozměry) výsledného obrobku.
- **Technologické informace:** určují technologii obrábění s ohledem na optimální řezné podmínky. Mezi technologické informace patří rychlost posuvu, velikost otáček vřetena či jeho směr, rychlost posuvu nebo hloubka řezu.
- **Pomocné informace:** obsahují ostatní informace potřebné pro výrobu obrobku na CNC stroji, jsou zde zahrnuty informace o zapnutí a vypnutí nebo chlazení vřetene, výměně nástroje, konec podprogramu a konec celého programu.
- **Ostatní informace:** v kódu programu jsou důležité pro orientaci v programu jako např. čísla bloků nebo poznámky.

Programovací kód je členěn do bloků (vět, řádků), kde tyto bloky jsou tvořeny posloupností jednotlivých slov (příkazů) obsahujících významovou a adresovou část. [2]



Obr. 1.4: Stavba jednoho bloku G-kódu složeného z příkazů [2]

Slova každého bloku jsou uspořádána dle normy v pořadí: **N G X Y Z F S T M**. Každá tato adresa má určitý význam:

- **N:** označuje číslo bloku.
- **G:** v názvu doplněné také o dvě čísla, tvoří **G-funkci**, tato funkce má především vliv na dráhu nástroje. **G-funkce** jsou rozděleny do skupin podle toho, jaký druh pohybu vykonávají. Nejdůležitější jsou funkce např. G00, G01, G02 a G03, které budou popsány níže. Většina těchto **G-funkcí** je modální, což znamená, že po vyvolání této funkce si drží svůj vliv až do konce chodu programu nebo do nahrazení této funkce jinou funkcí ze stejné skupiny.
- **X, Y, Z:** zastupují adresy a jsou doplněny číselnou hodnotou dané souřadnice, čímž určují pohyb jednotlivých os.
- **F:** slouží k zadávání rychlosti posuvu. Nejčastěji používané jednotky jsou posuv za minutu.
- **S:** slouží pro zadávání otáček vřetene.
- **T:** reprezentuje nástrojovou funkci, kde je tato operace používána pouze pro výběr nástroje.
- **M:** zastupuje funkce ovlivňující většinou technologii, jako třeba zastavení programu, konec programu, zapnutí vřetena po nebo proti směru hodinových ručiček či výměnu nástroje. [2], [5]

1.4.2 Důležité G a M funkce

V následujících tabulkách jsou vyobrazeny jedny z nejpoužívanější G a M funkce, které „by měly“ být součástí každého CNC programu. V ojedinělých případech funkčnost programu závisí na jejich použití.

Funkce rychloposuvu využívá při vykonávání maximální přípustné hodnoty stroje. Proti tomu funkce G01 pracovní posuvy lineární interpolací vyčítá funkční rychlosti z hodnot zadaných v pracovním programu u písmene F. U kruhových interpolací jsou rychlosti vyčítány také z programu.

Důležité funkce jsou funkce G54-G59. Kdy při použití těchto funkcí jsme schopni měnit souřadnice pracovního prostoru či stanovit nulový bod obrobku.

Tab. 1.1: Často používané g-funkce [1]

Funkce	Název	Popis
G00	Lineární interpolace	Rychloposuv
G01		Pracovní posuvy
G02	Kruhová interpolace	Ve směru hod. ručiček
G03		Proti směru hod. ručiček
G17-G19	Určení pracovní roviny	17: X-Y, 18: Z-X, 19: Y-Z
G33	Řezání závitů	Určení hloubky třísky
G40-42	Korekce	Ovládání korekcí rádiusů (zapnutí, vypnutí)
G53	Zrušení posunu nulového bodu	Použití když za posledním pohybem pokračuje pohyb v původních základních souřadnicích.
G54-G59	Změny nulového bodu	Absolutní posun nulového bodu.
G98	Reference	Nájezd stroje do referenčního bodu.

Z následující tabulky je nutné vyzdvihnout funkci M30. Bez použití této funkce s největší pravděpodobností dojde k nevykonání programu a to z důvodu nemožnosti interpolátoru definovat konec programu. Tzn. řídicí systém si bude myslet, že program vykonal, avšak fyzicky nebude žádný kód vykonán. [1], [3], [5]

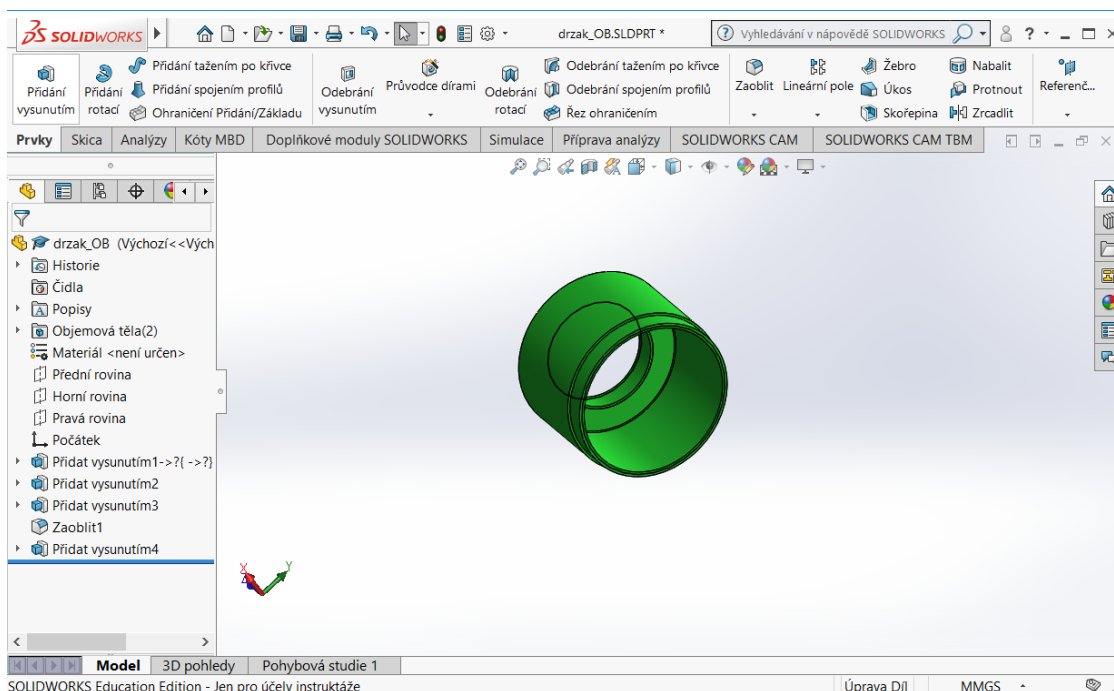
Tab. 1.2: Často používané m-funkce [1]

Funkce	Název	Popis
M00	Zastavení stroje	Po stisknutí tlačítka Start může program pokračovat. Vhodné např. pro odstranění třísek / špon.
M03	Otáčky vřetene	Ve směru hodinových ručiček.
M04		Proti směru hodinových ručiček.
M05	Zastavení vřetene	Vhodné použití při výměně nástroje či jeho korekce.
M06	Výměna nástroje	Používá se často při ruční výměně nástroje.
M07, M08	Zapnutí chlazení	M07 chlazení č.1, M08 chlazení č.2.
M09	Vypnutí chlazení	Obsluha čerpadla chlazení, mazání.
M30	Konec programu	Návrat na začátek hlavního programu.

1.4.3 Programy pro generaci kódu

Dříve se kód pro CNC stroje programoval ručně přímo do textového souboru. Dnes již existují modernější metody, jako jsou 3D editory. CAD/CAM software je počítačový software s integrovanou podporou konstrukce a výroby součástí. Název je tvořen zkratkou anglických názvů Computer-Aided Design a Computer-Aided Manufacture. V prostředí těchto softwarů je pomocí modelování vytvořen požadovaný obrobek, jehož výstupem bývá nejčastěji výkresová dokumentace,

ze které se generuje základní NC kód. Výstupem však nemusí být pouze NC kód, ale také výkresová dokumentace, data o využití materiálu a nástrojů. [4], [6]



Obr. 1.5: Příklad vývojového prostředí Solidworks

Mezi dnešní používané CAD/CAM systémy patří například Powermill či FutureCAM. Často bývají tyto programy požívány samostatně jako CAD, například Solidworks viz Obr. 1.5, nebo CAM, například Solidworks CAM. Dnes se také již používají CNC simulátory pro vizualizaci a výuku programování CNC.

1.5 Popis funkce řídicích systémů CNC

Základním úkolem CNC je tedy řízení pohybu nástroje v okolí obrobku, přičemž instrukce o dráze a rychlosti jsou stanoveny v programu. Zadané informace z programu a zpracované korekce jsou v řídicím systému použity jako logické funkce. Tyto funkce se jako řídicí povely dostanou do takzvaného adaptivního zařízení stroje, kde jsou propojeny se zpětnými hlášeními a následně převedeny na řídicí povely. Největší část dnešního řízení je realizována pomocí PLC (programovatelný logický počítač). Hlavními úkoly PLC jsou řízení, kontrola mechanických částí, reagovat na funkce ovládacího panelu, reagovat na snímané veličiny a práce s procesními daty. [7]

1.5.1 Ovládací režimy

Při obsluze stroje je možnost se setkat s více druhy ovládacích režimů stroje. Výběr těchto režimů je většinou volitelná. Řídicí systémy mají režimy obvykle:

- **Manual** – ruční provoz, slouží k přestavení nástroje nebo měřicího zařízení do požadované polohy, najíždění k obrobku, rozběh otáček, apod.
- **Auto** – automatický provoz, plynulé provádění programu. Stroj čte bloky programu a zpracovává je po sobě automaticky.
- **Blok po Bloku** – stroj pracuje v takovém poloautomatickém režimu, kdy se po bloku programu zastaví a po dalším startu čte následující blok.
- **Nastavení** – volba rychlostí posuvů a otáček, obvykle 0-100%.
- **Tool memory** – paměť nástrojových dat, umožňuje pracovat s daty o nástrojích, včetně korekcí.
- **Editace programu** – program pro obrábění je nahrán do editoru, kde je možná jeho editace, popřípadě oprava.
- **Diagnostika** – oznamuje, lokalizuje a diagnostikuje závady. [1], [4]

1.5.2 Současně dostupné řídicí systémy

V dnešní době je na trhu spousta výrobců, od malých firem až po korporátní společnosti, zabývajících se mechanickou konstrukcí či řídicími systémy pro obráběcí stroje. Navrch však mají výrobci slučující obě tyto odvětví dohromady.

Siemens Sinumerik

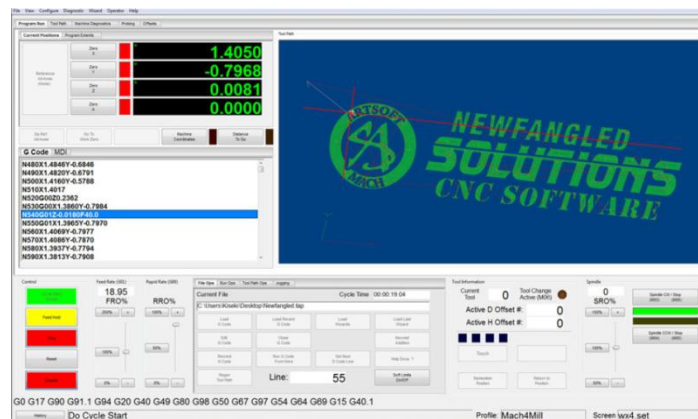
Nadnárodní společnost Siemens už nějaká ta léta působí na automatizační scéně. Má obrovské portfolio od malých komponent do rozvaděčů, přes PLC až po obrovské projekty. Na obrázku je možnost vidět panel integrující do sebe ovládací prvky a vizualizaci na obrazovce. Sinumerik 840 je otevřený CNC koncept pro modulární stroje. Obr. 1.6. [8]



Obr. 1.6: Sinumerik 840 [8]

Mach4

Mach4 je software pro CNC řízení obráběcího stroje. Od základu je napsán tak, aby byl rozšiřitelný, flexibilní a jednoduchý na používání. Dokáže také zpracovávat velké programy. Mach4 byl vyvinut jako evoluce svého předchůdce Mach3, který byl určen spíše pro hobby účely. Obr. 1.7 znázorňuje rozhraní softwaru Mach4. [9]



Obr. 1.7: Rozhraní softwaru Mach4 [9]

Beckhoff

Beckhoff je Německá společnost fungující na poli průmyslové automatizace. Zaměřuje se jak na software, přesněji řečeno TwinCAT, tak i na hardware a to od kabelů, svorek a I/O karet přes řídicí karty a frekvenční měniče až po průmyslové počítače a ovládací panely. CNC ovládací set je na obrázku 1.8. [10]



Obr. 1.8: CNC ovládací set [10]

2 ANALÝZA VÝCHOZÍHO STAVU FRÉZKY OPTIMUM VARIO BF 20L

Tato pasáž obsahuje výchozí popis stavu frézky doplněné o technické informace.

Vrtačko-frézka Optimum Vario BF 20L je vyrobena firmou Optimum Maschinen Germany GmbH. Frézka je vyobrazena na obrázku č. 2.1. Jedná se o velmi přesnou a velice stabilní vrtačko-frézku pro rychlou práci. Písmeno „L“ na konci názvu naznačuje, že jde o verzi vrtačkofrézky s větším křížovým stolem. Na stroji lze také plynule regulovat velikost otáček a směr otáčení. Velikost otáček se zobrazuje na display. [11]

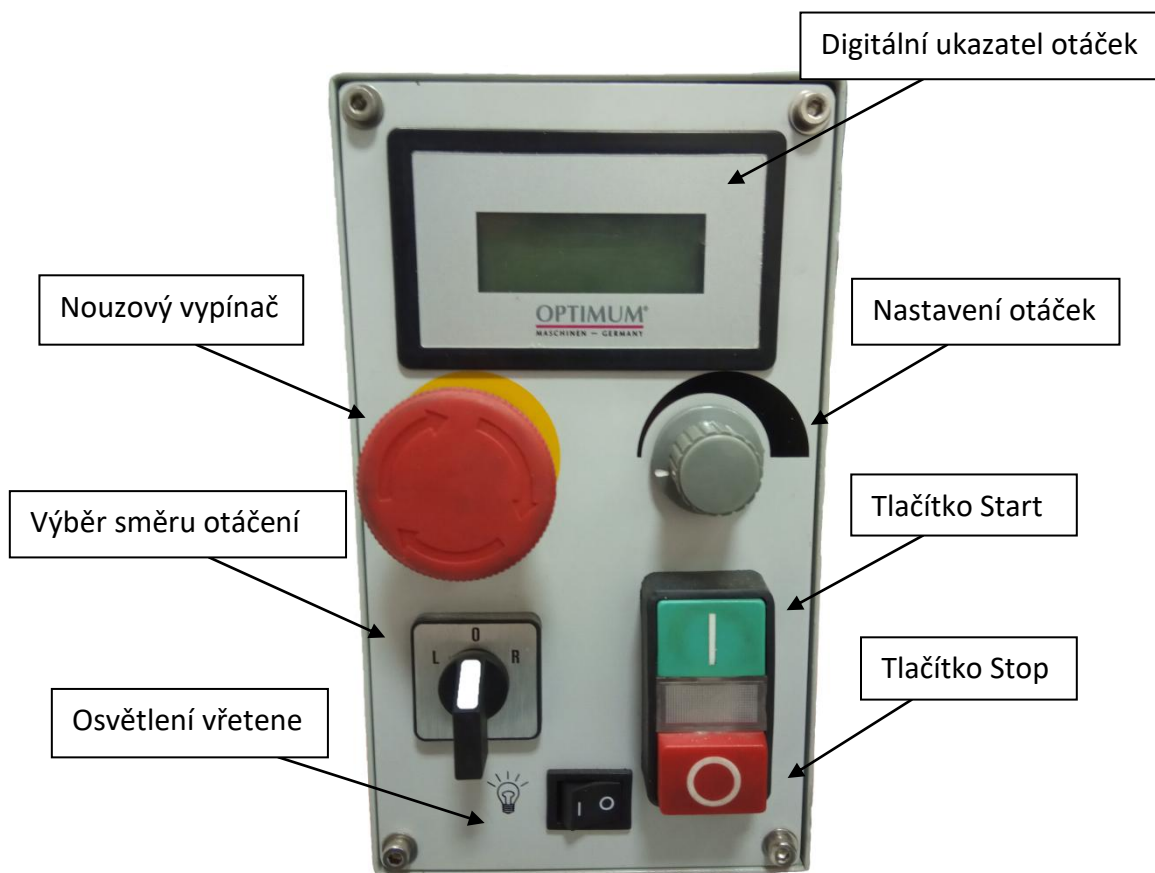


Obr. 2.1: Optimum Vario BF 20L [12]

2.1 Popis ovládání

Jednotlivé osy frézky jsou ovládaný pomocí ovládacích kol s odmontovatelnou rukojetí. Pro lepší orientaci míry velikosti pohybu jsou k osám připojeny pravítka.

Pro osu X, ve směru doprava a doleva jsou použity dvě ovládací kola, jedna na každé straně křížového stolu. Osa Y je myšlena ve směru dozadu a dopředu, kde je ovládací kolo namontováno zepředu do ložiskového pouzdra pro zachycení jak axiálního tak i radiálního pohybu šroubu stolu. Pohyb nahoru a dolů, tzv. pohyb v ose vřetene, je označen jako osa Z.



Obr. 2.2: Ovládací panel vřetene

Na obrázku 2.2 je vyobrazeno ovládání vřetene. Popis jednotlivých částí:

- **Nouzový vypínač:** Odpojení napájení vřetene.
- **Výběr směru otáčení:** L – proti směru, R – po směru hodinových ručiček.
- **Osvětlení vřetene:** Vypínač integrovaného osvětlení za vřetenem.

- **Digitální ukazatel otáček:** Zobrazení rychlosti vřetene v ot/min.
- **Nastavení otáček:** Potenciometr pro regulaci otáček.
- **Tlačítko Start:** Zapnutí vřetene.
- **Tlačítko Stop:** Vypnutí vřetene.

2.2 Technické parametry

V tabulce 2.1 je vypsán výčet důležitých technických parametrů frézky.

Důležitou poznámkou je také fakt, že pohyb os X a Y zajišťuje křížový stůl, jenž vlastně invertuje pohyb obrobku vůči nástroji, bereme-li v potaz, že základní CNC má pevný stůl s upevněným obrobkem a hýbe se pouze vřeteno. V tabulce jsou také zapsány stoupání jednotlivých trapézových šroubů. Tyto hodnoty nám následně poslouží pro výpočet správného převodu na hřídel motoru.

Tab. 2.1: Technické parametry frézky [11]

Technický údaj	Velikost	Jednotka
Motor vřetene	230 / 50 / 850	[V / Hz / W]
Hmotnost	103	[kg]
Rozměr – výška	860	[mm]
Rozměr – hloubka	670	[mm]
Rozměr – šířka	550	[mm]
Délka stolu	700	[mm]
Šířka stolu	180	[mm]
Rozsah posuvu v ose X	480	[mm]
Rozsah posuvu v ose Y	175	[mm]
Rozsah posuvu v ose Z	370	[mm]
Kužel vřetene	MK 2 / M10	[-]
Převodový stupeň „pomalu“	cca 50 – 1400	[1/min]
Převodový stupeň „rychle“	cca 100 – 3000	[1/min]
Hluk max.	78	[dB]
Stoupání trapézového šroubu os X a Y	2	[mm]
Stoupání trapézového šroubu osy Z	4	[mm]

3 HARDWARE

Obsahem kapitoly hardware je soubor možných mechanických prvků při přestavbě či stavbě CNC stroje a to zejména při realizaci jeho řízení. Teoretický popis odkazující se na jednotlivé prvky a dále také popis konkrétního použitého hardwaru s důrazem na hlavní aspekty a jejich vlastnosti.

Elektronika CNC strojů se skládá z následujících tří částí:

- **Vstupní obvody** - napájení, vstupy od čidel a ostatních měřicích prvků
- **Řídicí obvody** - na základě vstupů a programu ovládá výstupy
- **Výstupní obvody** - akční členy, např. motory

3.1 Napájení

Nedílnou součástí každého elektrického zařízení je jeho napájecí obvod. Vstupním napětím poté bude síťové napětí (400/230V, 50Hz).

Vnitřní obvody poté mohou obsahovat hlavní vypínač a určité jisticí prvky pro ochranu samotného zařízení. Elektronika řízení je nejčastěji napájena 24V stejnosměrného napětí a z tohoto důvodu je nutné, aby vstupní obvody obsahovaly také externí zdroj, který nám toto napětí vytvoří. Výkon zdroje by měl být volen s ohledem na odebíraný proud zařízení.

3.1.1 Použité napájecí prvky

- **Hlavní vypínač**

Použit je vačkový spínač od české firmy Obzor. Jeho nominální proud je 16 A. K vypínači je možné připojit jak jednofázové, tak i třífázové napětí. Vypínač je znázorněn na obrázku 3.1



Obr. 3.1: Hlavní vypínač [13]

- **Hlavní zdroj**

Řídicí systém, společně s jeho moduly, popřípadě i fyzické vstupy a výstupy, vyžadují napájení 24V.

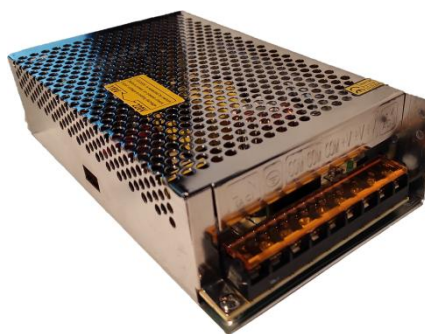
Zdroj Mean Well, viz Obr. 3.2, s výstupem 480W, vysokou odolností a možností montáže na DIN lištu představuje ideálního kandidáta na použití. Jeho efektivita dosahuje 92% a s proudem do 20A dokáže napájet všechny požadované prvky.



Obr. 3.2: Hlavní zdroj MW [14]

- **Zdroj pro napájení os**

V závislosti na ovládacích kartách, které disponují řízením motorů až na 48V byly využity tři spínané zdroje, pro napájení ovládání každého motoru jeden, typu CarSPA HS-250-48. Zdroj se nachází na obrázku č. 3.3.



Obr. 3.3: Spínaní zdroj CarSPA

- **Jištění**

V plánu je také doplnění o jištění jako případnou ochranu před elektrickým proudem. Průmyslový rozvaděč bude doplněn o řadovou pojistkovou svorku na DIN, viz obrázek 3.4. Tyto pouzdra budou doplněny odpovídajícím pojistkám pro maximální možné napájení.



Obr. 3.4: Pojistkové pouzdro [15]

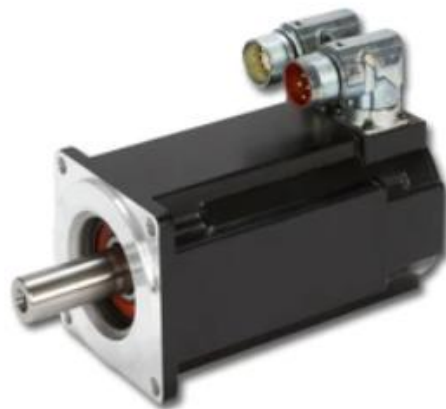
3.2 Elektrické pohony

Elektrické pohony jsou hlavními pilíři při stavbě CNC stroje. Na jejich volbě závisí správná funkce stroje. Nejběžněji používané pohony jsou servomotory a krokové motory. Pro specifikaci správného motoru je nutné znát požadovaný moment, rychlost, zrychlení a setrvačnou hmotnost zátěže. Důležitými parametry jsou také velikost a hlavně cena.

Lehké porovnání na úvod. Servomotory jsou vhodné pro aplikace, kde je požadována vyšší rychlost, akcelerace a přesnost. Naproti tomu krokové motory mnohem levnější a kompaktnější, jejich výhodou je také vyšší moment při nízkých otáčkách. [16]

3.2.1 Servomotor

Servomotor je synchronní, asynchronní či stejnosměrný motor, u kterého lze, na rozdíl od ostatních motorů, nastavit přesnou polohu natočení osy, viz obr. 3.5. Tyto motory musí být řízeny zpětnovazebně a jsou regulovány buď proudově, rychlostně nebo polohově. Díky své přesnosti a spolehlivosti jsou většinou používány u CNC strojů, čtecí hlavy pevných disků nebo v případě modelářského použití.



Obr. 3.5: Servomotor [16]

Asynchronní servomotory jsou žádoucí v aplikacích, kde je nutné pohybovat s velkou setrvačností hmoty. Oproti tomu synchronní servomotory jsou volbou v ostatních případech, s doplněním o vhodný frekvenční měnič. [17]

3.2.2 Krokový motor

Krokový motor je synchronní motor, obrázek 3.6, skládá se ze statoru a rotoru. Ovládání je prováděno přiváděním stejnosměrného napětí na jednotlivé cívky statoru. Rychlost rotoru je rovna točivému magnetickému poli ve statoru. Název odvozen od pohybu motoru krok po kroku. Typická velikost kroku je rovna $1,8^\circ$ při 200 krocích na otáčku. Při mikrokrokování lze ještě každý krok rozdělit až na 64 kroků.



Obr. 3.6: Krokový motor [16]

Stator je vnější část motoru. Je vyroben z ocelových lamel s drážkami, ve kterých se nachází měděné vinutí. Tyto lamely vytvářejí několik dvojic cívek, které mohou být zapojeny více způsoby. Rotor je točivá část motoru s vyniklými póly vyrobená z magnetického materiálu. [18], [19]

Tab. 3.1: Výhody a nevýhody krokových motorů [19]

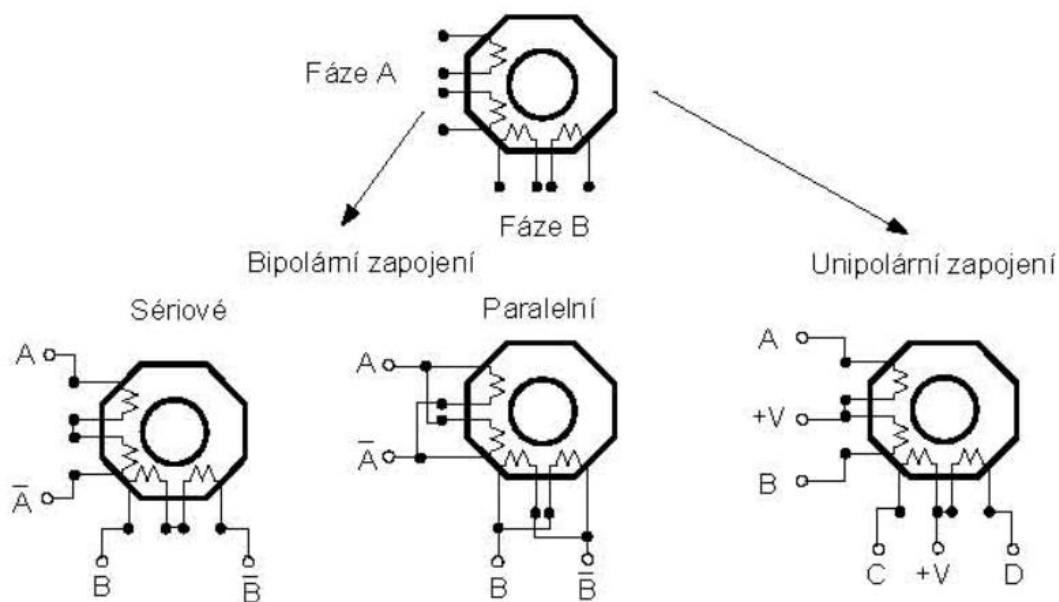
Výhody	Nevýhody
<ul style="list-style-type: none">• jednoduchý a laciný• zcela digitální motor• bez akumulace úhlové chyby• bezkartáčový - minimální údržba• stejná poloha i při nulové rychlosti – stacionární moment• vysoký moment při nízkých otáčkách• nelze mechanicky přetížit• bez potřeby zpětné vazby• pohyb oběma směry• kompatibilní od různých výrobců	<ul style="list-style-type: none">• při přetížení ztratí pozici• extrémně citlivý při přetížení• ztrácí moment při rostoucích otáčkách• bez zpětné informace o skutečné poloze - lze použít enkodér• limitovaná dynamika• je poměrně hlučný - zvláště při rezonanci

Způsob zapojení

Zapojení krokových motorů je popsáno na obr. 3.7 a patří mezi ně tyto typy:

- **Unipolární** – Je zde vyveden střed cívky, proud může protékat pouze jedním směrem a v jednu chvíli je sepnutá pouze jedna cívka. Zapojení má nižší odběr a menší moment. Výhodou však je jednoduchost jeho zapojení.
- **Bipolární sériové** – Elektrický proud zde prochází zároveň protilehlými cívkami. Sériové zapojení cívek.
- **Bipolární paralelní** – Elektrický proud zde prochází zároveň protilehlými cívkami. Paralelní zapojení cívek.

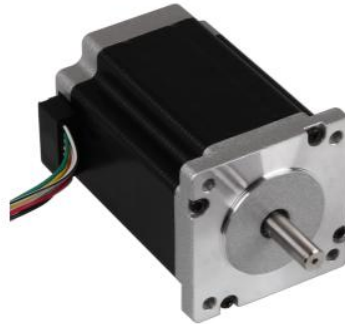
Pokud se podíváme detailněji na bipolární zapojení, tak každá varianta skýtá určité výhody a nevýhody. Sériové zapojení má vyšší klidový moment, který však rychle klesá s rostoucími otáčkami. Paralelní zapojení je schopno si s rostoucími otáčkami udržet vyšší točivý moment. Toto vychází z podstaty, kdy paralelní zapojení má čtvrtinový odpor a indukčnost oproti sériovému. Což ale také vede k vyššímu proudu, většímu generovanému teplu a ruku v ruce ke snížení počtu pracovních cyklů cca o 50%. [20]



Obr. 3.7: Varianty zapojení krokových motorů [21]

3.2.3 Použité pohony

Jako elektrický pohon je vybrán krokový motor NEMA23-1 60STH86-30008D v počtu 3ks, viz obr 3.8. Tyto motory disponují 200 kroky na otáčku. Pohyb je řízen čtyřmi fázemi. Součástí motorů je také prodloužená hřídel, která umožňuje připojení enkodéru. [22]



Obr. 3.8: Použitý krokový motor [22]

- IRC

Motor je doplněn o inkrementální rotační enkodér typu S3806. [23]

3.3 Řídicí systém

Řídicí systém CNC je srdce i mozek CNC obráběcího stroje. Při obrábění budete závislí na jeho výkonu, rychlosti zpracování dat, spolehlivosti a snadnosti obsluhy. Důležitou otázkou je, co by měl CNC řídicí systém umět a pro jaký typ výroby ho požadujete. [24]

Na trhu je široká škála řídicích systémů. Spousta z nich má vlastní elektroniku řízení navrženou a vytvořenou pro přesně daný typ stroje nebo řadu strojů. Pro tyto účely jsou také často používány programovatelné logické automaty.

Programovatelné logické automaty jsou však také používány při stavbě či retrofit hobby obráběcích strojů. Mezi jiné patří do skupiny hobby CNC řídicích systémů také Arduino nebo Raspberry PI, jež disponují nižší cenou.

3.3.1 Programovatelný logický automat

Programovatelný logický automat, ve zkratce řečeno PLC (Programmable Logic Controller), je dnes relativně malý počítač v průmyslovém provedení. Tento počítač je také řízený mikroprocesorem s vlastním operačním systémem, na rozdíl od normálních počítačů jsou tyto systémy uzpůsobeny pro potřeby řešení automatizačních úloh v reálném čase, s co nejmenší dobou odezvy, jako například řízení výrobních linek či strojů v továrně. Odlišnost PLC automatů na rozdíl od běžných počítačů nespočívá jen v cyklickém zpracování programu, ale také

v přímém napojení periférií na technologické procesy. Signály z řízeného procesu jsou tedy přivedeny rovnou na vstupní periférie, kterými je PLC vybaveno pro komunikaci s okolím. Na druhé straně jsou zde i výstupní periférie pro ovládání akčních prvků řízeného procesu. Převážnou část periférií zastupují digitální vstupy (DI), digitální výstupy (DO) a analogové vstupy (AI), analogové výstupy (AO). S postupem času a rozvojem automatizace se začaly používat další periferní jednotky, jako například komunikační moduly, safety moduly či vybavení PLC jednotkou HMI (human machine interface), display.

Podle velikosti jsou PLC rozdělena na malé, střední a velké. Velikostí nejsou myšleny jen konstrukční rozměry, ale také výkon, který se odvíjí od velikosti uživatelské paměti, rychlosti zpracování instrukcí, počtu připojitelných vstupů a výstupů a dalších parametrů. Rychlost zpracování vstupních a výstupních dat ve značné míře také ovlivňuje integrované rozhraní PLC.

Z hlediska konstrukce mohou být programovatelné automaty kompaktní nebo modulární. Kompaktní PLC obsahuje integrované CPU, digitální a analogové vstupy a výstupy a základní podporu komunikace, s malou možností rozšiřitelnosti. U modulárního PLC jsou jednotlivé komponenty rozděleny do modulů. Celé PLC se potom skládá z modulu zdroje, CPU, komunikačního modulu, téměř libovolného počtu vstupních a výstupních modulů a dalších možných modulů. Modulární PLC je dále možno snadno rozšiřovat s ohledem na limity PLC. [25], [26]

3.3.2 Použité řízení

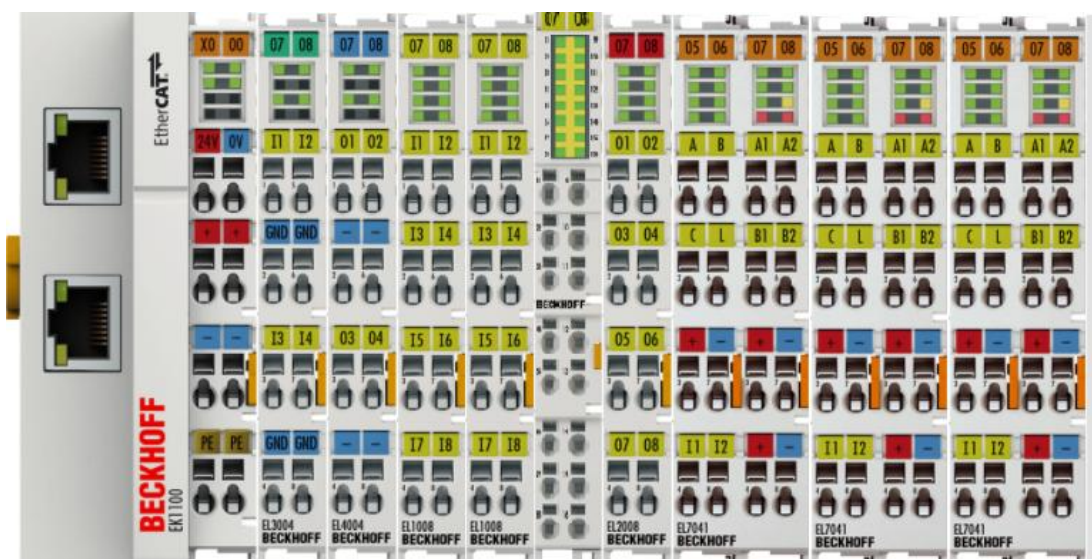
- **C6015-0010**

Ultra-kompaktní průmyslový počítač. O jeho výkon se stará čtyřjádrový Intel Atom 1,9 GHz a operační paměť 4 GB, společně s integrovaným SSD diskem. Obsluhu komponent v reálném čase zde vykonává softwarové PLC, které běží na stejné úrovni jako Win10. Obr 3.9. [10]



Obr. 3.9: IPC C6015-0010 [10]

Obsluha vstupů a výstupů je zprostředkovány pomocí modulárních I/O terminálů připojených skrz coupler, viz obr. 3.10.



Obr. 3.10: Použité I/O karty

- **EK1100** – EtherCAT Coupler – komunikační terminál
 - provádí výměnu dat mezi EtherCAT průmyslovou sběrnici a Ethernetem
 - napájení 24 V je nutné přivést jak pro EtherCAT (24 V, 0 V), tak pro silovou část (+, -)
- **EL3004** – 4-kanálový analogový vstupní terminál (12 bit)
 - rozsah vstupního signálu -10..+10 V
- **EL4004** – 4-kanálový analogový výstupní terminál (12 bit)
 - rozsah výstupního signálu - 0..+10 V
- **EL1008 (2x)** – 8-kanálový digitální vstupní terminál
 - nominální napětí 24 V (log. 1 – 11..30 V, log. 0 - -3..+5V)
- **EL1809** – 16-kanálový digitální vstupní terminál
 - nominální napětí 24 V (log. 1 – 11..30 V, log. 0 - -3..+5V)
- **EL2008** – 8-kanálový digitální výstupní terminál
 - nominální napětí 24 V (max. výstupní proud 0.5 A)
- **EL7041 (3x)** – terminál pro krokový motor s enkodérem
 - napájení ovládání motoru 48V
 - maximální výstupní proud 5A
 - napájení enkodéru 24V
- **EL9011** – krytka konce sběrnice [10]

3.4 Ostatní použité elektrické zařízení

3.4.1 MPG ovladač

Přesné ruční ovládání je zprostředkováno ručním ovladačem ISM1468, obr. 3.11.

Vlastnosti ovladače:

- Ruční ovládání 4 os
- MPG kodér 100 imp/ot
- IRC výstup (A, B)
- STOP tlačítko
- Tlačítko ON/OFF
- Indikace LED
- Přepínač volby os (OFF, X, Y, Z)
- Přepínač velikosti kroku (X1, X10, X100)
- Držák s magnety [27]



Obr. 3.11: Ruční ovladač ISM1468 [27]

3.4.2 Čidlo korekce délky nástroje

Jako senzor délky upnutého nástroje je zvolen jednoduchý typ TS32. Tento senzor je vybaven mikrospínačem a čidlem, které detekuje nástroj v případě jeho dotyku styčné plochy shora, viz obr. č. 3.12.

Při výměně nástroje senzor přesněji nastaví hloubku frézování. [28]



Obr. 3.123: Čidlo korekce délky nástroje TS32 [28]

3.4.3 Koncové čidla

Jako koncové čidla byly zvoleny čidla typu induktivního s PNP logikou (NO), viz obr. 3.13. Napájení senzorů je 24V, připojení je realizováno 3-žilovým kabelem (2-žily napájecí, 1-žila signální).

Čidla jsou vytvořeny z nerezové oceli o velikosti 8mm a detekční vzdáleností 3 mm. [29]



Obr. 3.13: Referenční čidla[29]

3.4.4 Dotyková obrazovka

Pro interakci člověk-stroj je zvolena dotyková obrazovka ASUS VT168H, znázorněna na obrázku 3.14.

Velikost displaye činí 15,6“ s rozlišením 1366 x 768 px. Obrazovka umožňuje až 10ti bodové ovládání a je kompatibilní s Windows 10.

Připojení obrazu je provedeno pomocí HDMI portu. Přenesení dotyku realizuje port micro-USB. [30]



Obr. 3.14: Dotyková obrazovka[30]

4 REALIZACE KONSTRUKCE A EL. ZAPOJENÍ

Kapitola praktické realizace je orientována na mechanickou konstrukci, ve smyslu dovybavení os frézky a vytvoření jejího virtuálního 3D modelu. Dále se zaměřuje na elektrické zapojení průmyslového rozvaděče, který je následně použit jako počítačové řízení 3-osé frézky. K rozvaděči jsou také připojeny vstupní a výstupní prvky, které jsou nastíněny již v předchozí kapitole. Elektrotechnické schéma rozvaděče je vytvořeno v programu ProfiCAD.

4.1 Mechanická rekonstrukce

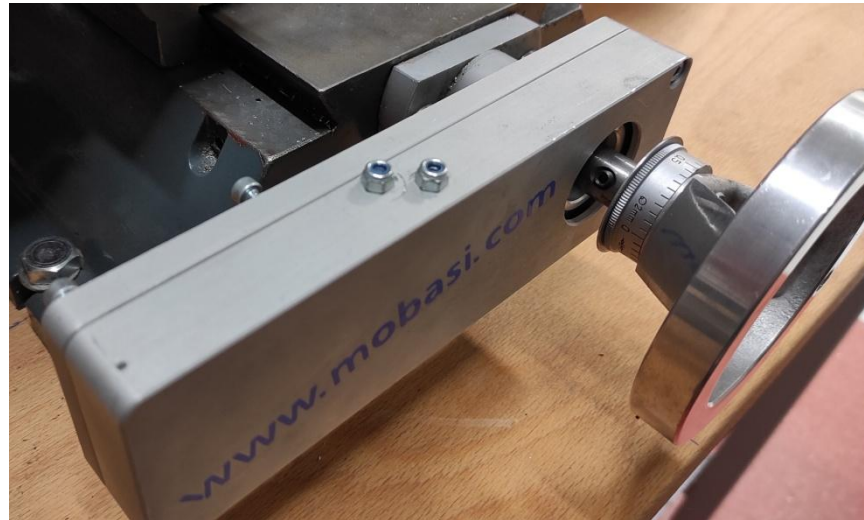
Jako retrofit frézky byla zvolena speciální souprava dílů. Tento kit od společnosti Mobasi obsahuje tři pod-soupravy, pro každou osu jednu, které obsahují pouzdro s přídatnými prvky pro uchycení motoru na určitou osu a dále také ozubený řemen s dvěma ozubenými řemenicemi, které slouží pro převod pohyb motoru na danou osu v daném poměru.

- **Osa X** – ve smyslu pohybu doprava, doleva. Při pohledu na původní frézku je zřejmé, že manuální ovládání je zde realizováno na obou stranách stolu. Proto pravá klika zůstane neměnná a levá je nahrazena převodovým krytem, který je zobrazen na obrázku 4.1. Do otvoru na obrázku se vloží motor NEMA-23 s nainstalovanou ozubenou řemenicí na své hřídeli.



Obr. 4.1: Retrofit osy X

- **Osa Y** – ve smyslu pohybu dopředu, dozadu. Přestavba na této ose byla nejkomplikovanější. Je nutné odmontovat kliku a pouzdro společně s doplňky naimplementovat na její místo. Motor se zde nachází rovnoběžně se základnou stolu. V případě že požadujete i manuální ovládání, je možné kliku zpět namontovat na pouzdro. Problém nastává ve chvíli, pokud máte starší frézku, zde je pro montáž kliky nutností prodloužit osu (nastavit trapézový šroub). Montáž společně s nastavenou hřídelí patrná na obr. 4.2.



Obr. 4.2: Montáž kitu osy Y

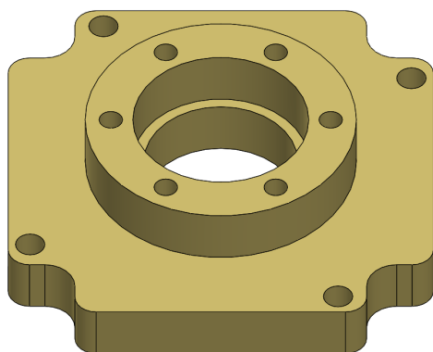
- **Osa Z** – ve smyslu pohybu nahoru, dolů. Zde se retrofit provedl odstraněním vrchní krytky trapézového šroubu a montáží kitu pro osu Z na toto místo. Kryt převodu osy Z se zásadně neliší od předchozích vyobrazených. Motor visí svisle dolů a manuální ovládání zůstává neměnné.

4.1.1 Připevnění motorů a enkodérů

Upevnění motorů k frézce se provede skrze příslušné otvory v pouzdrech.

Jelikož jsou zvoleny motory s prodlouženou hřídelí a na tuto hřídel má být připojen enkodér, je nutno enkodér s motorem vůči sobě zafixovat.

Tato fixace je provedena pomocí vytvořeného dílu, obrázek 4.3, a distančních sloupků připevněných mezi nástavec a enkodér. Nástavec je k motoru připevněn prodlouženými šrouby motoru. Celkové pospojování je patrné z obrázku 4.4.



Obr. 4.3: Nástavec motoru



Obr. 4.4: Fixace motoru a enkodéru

Převod rotačního pohybu z motoru na hřídel enkodéru je zajištěn pomocí hřídelové spojky.



Obr. 4.5: Celkový vzhled frézky

4.1.2 Celkový vzhled a upevnění frézky

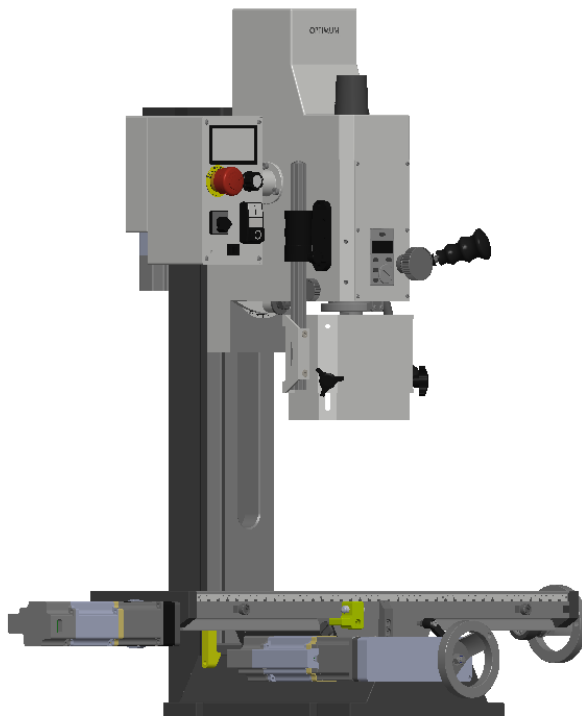
Frézka je upevněna k pojízdnému stolu, což usnadňuje její obsluhu a také dává prostor k manévrování, kdyby bylo nutné ji použít v jiných prostorách, viz obr. 4.5.

Rozvaděče s elektronikou a základním ovládáním jsou upevněny zespod stolu. Na frézku jsou také namontovány limitní snímače polohy s jejich držáky, popsány v kapitole 4.2.1.

Monitor pro ovládání frézky je upevněn na pohyblivé konzole, připojené k nosnému profilu kruhového tvaru. Na horizontu profilu je umístěn majáček signalizující zapnutí, chod a chybu stroje.

4.2 3D model frézky

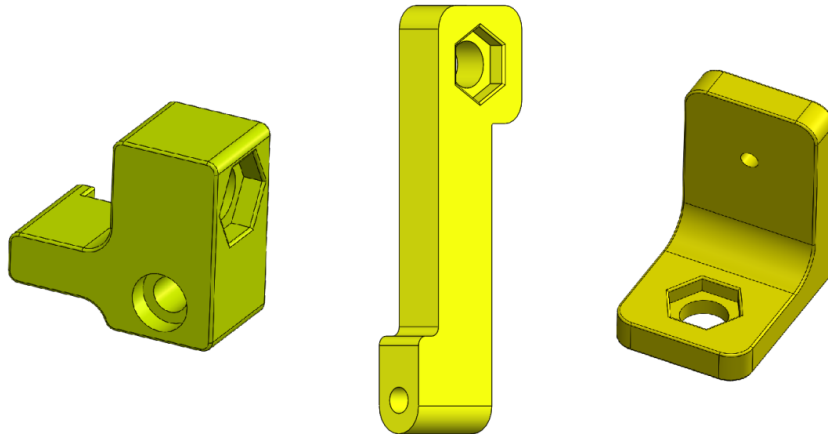
3D model retrofitu frézky je vytvořen v programu Solidworks. Základem je model požitého typu frézky z komunity GrabCAD [31]. V nynější podobě, viz obr. 4.6, se model skládá z 1100 dílů. Počet doplněných dílů je roven několika desítkám. Mezi hlavní doplněné díly patří soustavy pouzder os, motory s jejich nástavci a kryty a také elementy pro uchycení snímačů.



Obr. 4.6: 3D model retrofitu frézky

4.2.1 Držáky koncových snímačů

Limitní snímače jsou k frézce uchyceny pomocí namodelovaných držáků. Držáky jsou vyobrazeny na obr. 4.7, zleva náleží k osám X, Y a Z. Tyto díly jsou navrženy s ohledem na možnost uchycení na frézce a zároveň zásadně neomezí rozsah pohybu os. Držáky je většinou možné uchytit pouze v jednom bodě, proto jejich tvar po montáži zajišťuje aretaci na místě. Aretaci snímače zajišťuje otvor na matici.



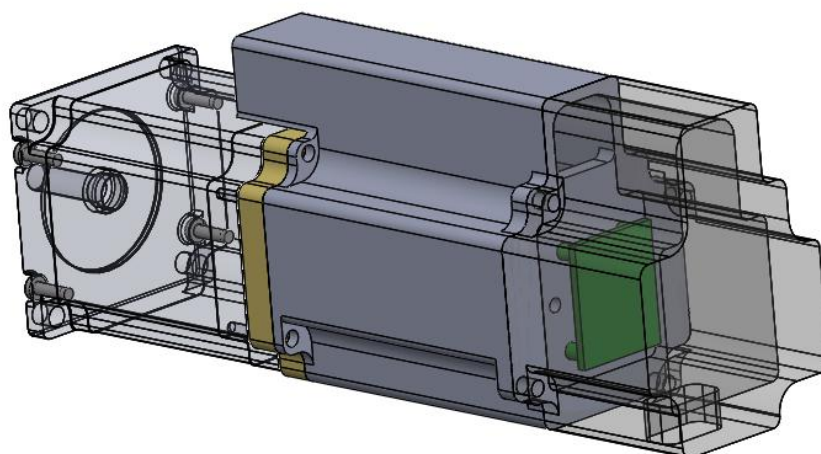
Obr. 4.7: Modely držáků snímačů X, Y, Z (zleva)

- **Osa X** – držák umístěn v přední části podstavce frézky, čidlo reaguje na zarážku křížového stolu vlevo.
- **Osa Y** – držák umístěn do boku v zadní části podstavce frézky, čidlo reaguje na samotný křížový stůl při posunu dozadu.
- **Osa Z** – držák umístěn k pouzdru kitu osy Z, čidlo reaguje při výsunu ústrojí vřetene do dostatečné výšky.

4.2.2 Vytvořený model krytu motoru

Spojení motorů a enkodérů bylo podmíněno výrobou krytů. A to jak z důvodu při ochraně před špony při frézování, tak z možnosti neopatrné obsluhy frézky.

Obrázek 4.8 popisuje model krytů s motorem. Zleva se jedná o motor (průhledný), nástavec (žlutý, již popsán dříve) a kryt, který se skládá ze dvou částí. Základní část krytu (modro-šedý) zajišťuje bezpečné ukrytí enkodéru. Vrchní část krytu (průhledný) slouží pro ochranu přepojení vodičů. Oba kryty jsou doplněny o tunel pro vedení kabeláže.

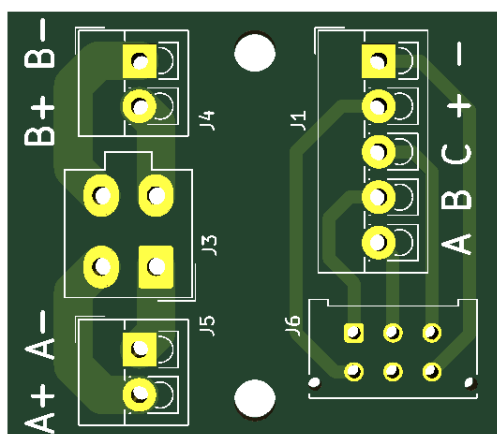


Obr. 4.8: Model motoru s krytem

4.2.3 Přepojení motorů

Přepojení vodičů motoru a enkodéru je prováděno na vytvořené desce plošných spojů (DPS), obr 4.9. Z vnější části jsou kabely připojeny do označených svorkovnic. Vnitřní připojení je realizováno konektory, aby bylo možné rozebrat kryt motoru a zároveň nepomíchat zapojení. Náhled na vytvořenou desku je na obrázku 4.10.

Deska je navržena co nejmenší, aby se vešla pod vrchní kryt, ale zároveň dostatečně dimenzována, aby odolala proudovému zatížení. Pro motor je použit konektor Mini-Fit, pro enkodér Micro-Fit.



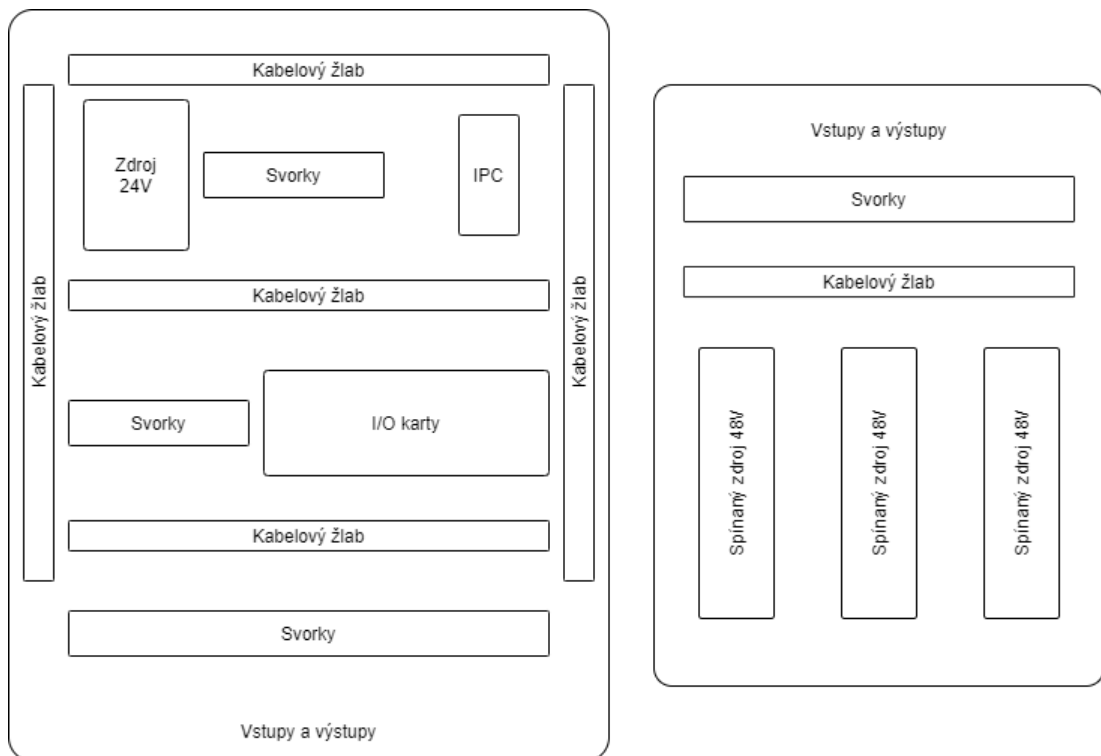
Obr. 4.9: DPS přepojení motoru



Obr.4.10: Osazená DPS na krytu

4.3 Elektrické zapojení

Elektrické zapojení všech komponent je umístěno ve dvou rozvaděčích. Navržené blokové rozložení komponent v každém rozvaděči je na obrázku 4.11. Hlavní rozvaděč obsahuje komponenty pro řízení. Vedlejší rozvaděč zastřešuje napájecí funkci. Rozvaděče jsou připojeny sobě v obráceném pořadí. Oba rozvaděče jsou pro ochranu proti přehřátí dovybaveny chladicími otvory (mřížky). Celkové elektrické schéma zapojení je vytvořeno v programu ProfiCAD (doplněno v příloze).



Obr. 4.11: Blokový návrh rozvaděčů (hlavní vlevo, vedlejší vpravo)

V rozvaděčích je zvolena konvence týkající se barvy kabelů:

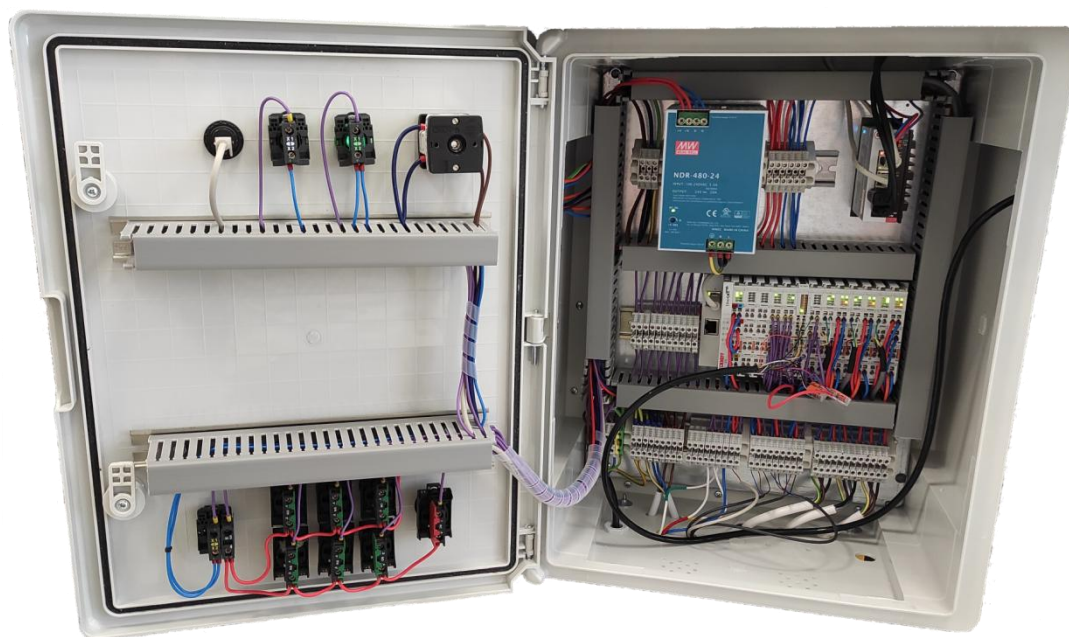
- **Hnědá** – fáze síťového napětí 230V
- **Tmavě modrá** – nulový potenciál síťového napětí
- **Zeleno-žlutá** – uzemnění
- **Červená** – silový rozvod 24V DC
- **Oranžová** – silový rozvod 48V DC
- **Modrá** – nulový potenciál vůči 24V, popřípadě 48V
- **Fialová** – signálový rozvod 24V DC

4.3.1 Hlavní rozvaděč

Hlavní rozvaděč má velikost 400x500x250 mm a nachází se v něm tyto prvky: zdroj 24V stejnosměrného napětí, průmyslový počítač (IPC) se softwarovým PLC, I/O karty připojené přes coupler a ethernetový kabel do průmyslového počítače.

Napájení do rozvaděče je přivedeno skrz svorky vlevo dole a poté přes hlavní vypínač, přes který je z externích zásuvek ze zadní strany rozvaděče, upevněné na din liště, napájen monitor a vřeteno. V rozvaděči jsou implementovány signalizační a ovládací prvky, pro nouzové ovládání, v podobě led kontrolky, tlačítek a přepínače. Doplněn je zde také ethernetový konektor.

Všechny vstupy a výstupy přivedené a vyvedené z a do rozvaděče jsou připojeny prostřednictvím svorek a to zejména kvůli zpřehlednění a ochraně rozvaděče. Výjimkou je MPG ovladač, který je připojen přímo do vstupního terminálu a na jeho přívodu je proti vytržení použita ochranná vložka. Zapojení rozvaděč je znázorněno na obrázku 4.12.

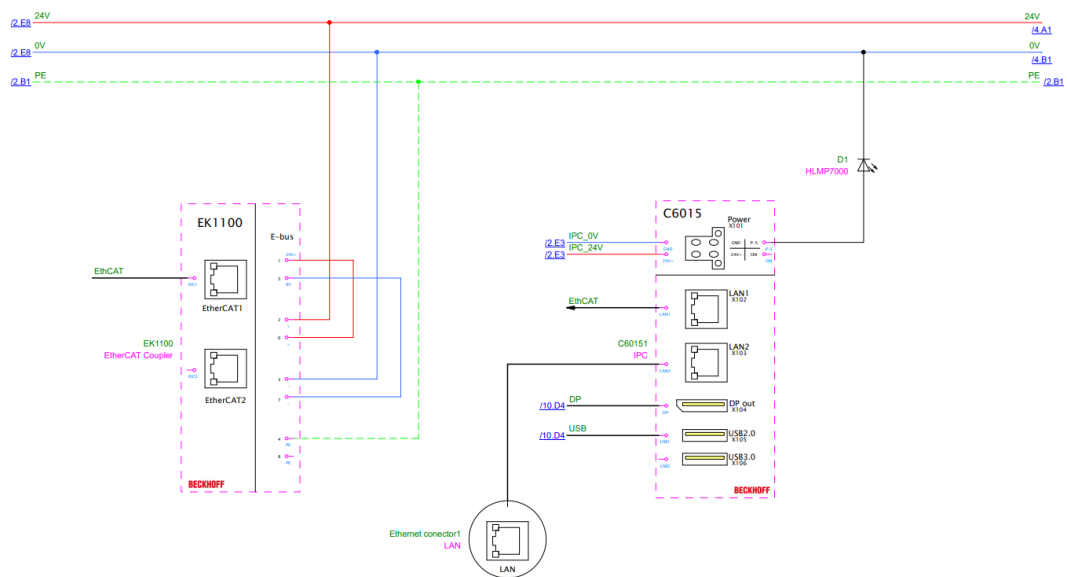


Obr. 4.12: Zapojení hlavního rozvaděče

Svorky umístěné vlevo od zdroje slouží k převedení síťového napájení od vypínače ke spínaným zdrojům do vedlejšího rozvaděče. Připojení dotykové obrazovky je vedeno horní stěnou rozvaděče.

Jednotlivé komponenty rozvaděče jsou detailněji popsány v kapitole 3.

4.3.2 Připojení IPC a coupleru



Obr. 4.13: Připojení IPC a EtherCAT coupleru

Elektrické schéma zapojení průmyslového počítače a EtherCAT coupleru je na obrázku 4.13. Na terminál EK1100 je přivedeno napájení na silovou sběrnici, ze které jsou poté napájeny další terminály, a také je zde přepojeno napájení na datovou sběrnici, tedy EtherCAT.

Propojení IPC a EK1100 je realizováno ethernetový kabelem na x102. Napájení 24V je přivedeno přes konektor, spolu se signalizací stavu zapnutého IPC vyvedené z pinu „P-S“. Obrazovka je připojena kabely DP->HDMI a USB->microUSB. Ethernetový konektor je připojen do slotu x103.

4.3.3 Připojení vstupů

- **Analogové vstupy**

Napěťová smyčka je nachystána na analogové kartě EL3004, piny 1 (I1) a 2 (GND). Určena je pro odečítání rychlosti (otáček) vřetene.

- **Digitální vstupy**

Digitální vstupy jsou připojeny pomocí tří terminálů. Popis funkcí dvou terminálů EL1008 je popsán v tabulce 4.1. Připojení ručního ovladače do vstupního terminálu EL1809 je popsáno v kapitole 4.3.6.

Tab. 4.1: Seznam digitální vstupů

Terminál	Vstup	Pin	Typ	Funkce
Term 2	I1	1	přepínač	Výběr ovládání pohotovostní – ostatní
	I2	5	tlačítko	Aktivace (enable) motorů
	I3	2	tlačítko	Posuv v ose X směr mínus
	I4	6	tlačítko	Posuv v ose X směr plus
	I5	3	tlačítko	Posuv v ose Y směr mínus
	I6	7	tlačítko	Posuv v ose Y směr plus
	I7	4	tlačítko	Posuv v ose Z směr mínus
	I8	8	tlačítko	Posuv v ose Z směr plus
Term 3	I1	1	senzor	Limitní indukční čidlo – osa X
	I2	8	senzor	Limitní indukční čidlo – osa Y
	I3	2	senzor	Limitní indukční čidlo – osa Z
	I4	6	senzor	Senzor délky nástroje

4.3.4 Připojení výstupů

- **Digitální výstupy**

Digitální výstupy jsou připojeny pomocí jednoho terminálu EL2008. Popis funkcí jednotlivých výstupů je uveden v tabulce 4.2.

Tab. 4.2: Seznam digitálních výstupů

Terminál	Výstup	Pin	Typ	Funkce
Term 4	O1	1	MPG_led	Výstup na led ručního ovladače - enable
	O2	5	led	Ledka signalizující error
	O3	2	led	Ledka signalizující aktivaci motorů
	O4	6	majáček	Error
	O5	3	vřeteno	Zapnutí vřetene po směru hod. r.
	O6	7	vřeteno	Zapnutí vřetene proti směru hod. r.
	O7	4	majáček	Stroj připraven na ovládání
	O8	8	majáček	Běh automatického programu

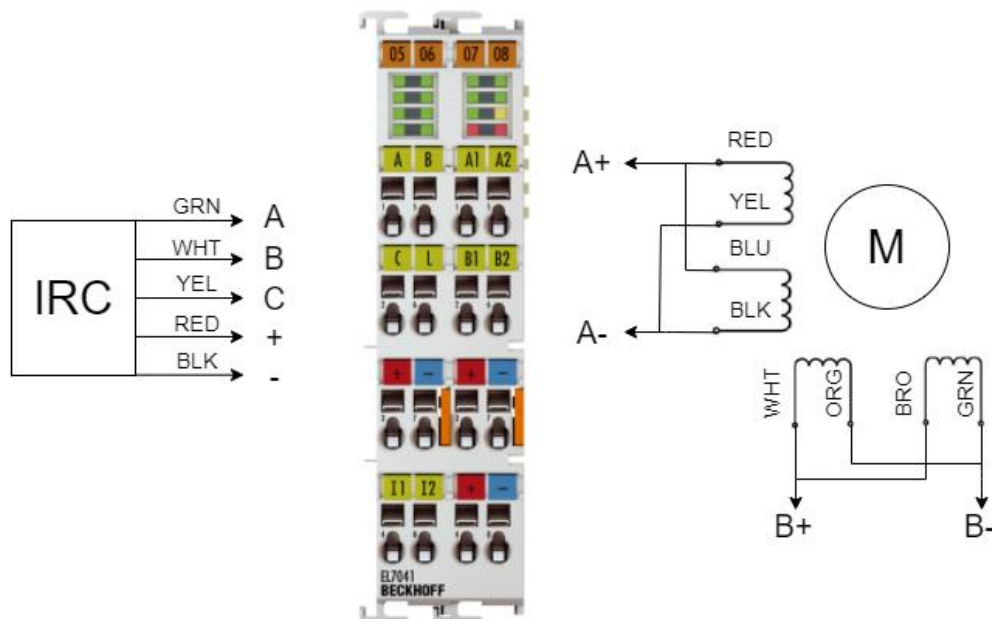
- **Analogové výstupy**

Napěťová smyčka je nachystána na analogové kartě EL4004, piny 1 (O1) a 2 (-).
Určena je pro regulaci otáček vřetene.

4.3.5 Připojení motoru ke kartě EL7041

Jelikož byl zvolen motor s osmi vývody a karta umožňuje připojit 4 řídicí výstupy, bylo pro motory vybráno paralelní bipolární zapojení, kdy by měl motor odebírat 4,2 A a karta je schopna dodávat 5 A. Připojení je na obrázku 4.14.

Terminál se skládá ze dvou částí, pravá část plní funkci ovládání motorů a levá část umožňuje připojení a napájení enkodéru. Enkodér je napájen skrze silové kontakty 24V. Avšak pro napájení motorů je nutné přivést externích 48V ze spínaných zdrojů ve vedlejší rozvaděči. Označení signálů terminálu do motorů A1, A2, B1, B2 je ekvivalentní označení vodičů motorů A+, A-, B+, B-.



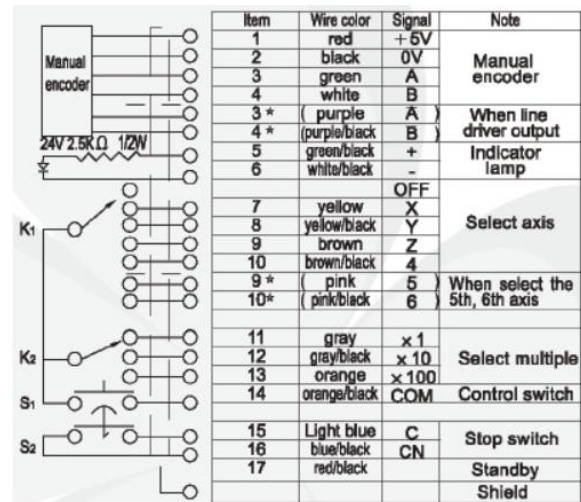
Obr. 4.14: Připojení terminálu krokového motoru EL7041

4.3.6 Připojení ručního ovladače

Ruční ovladač je nutné připojit podle obrázku 4.15. Stop tlačítko je zde realizováno pouze jako rozpínací kontakt použitelný dle libosti.

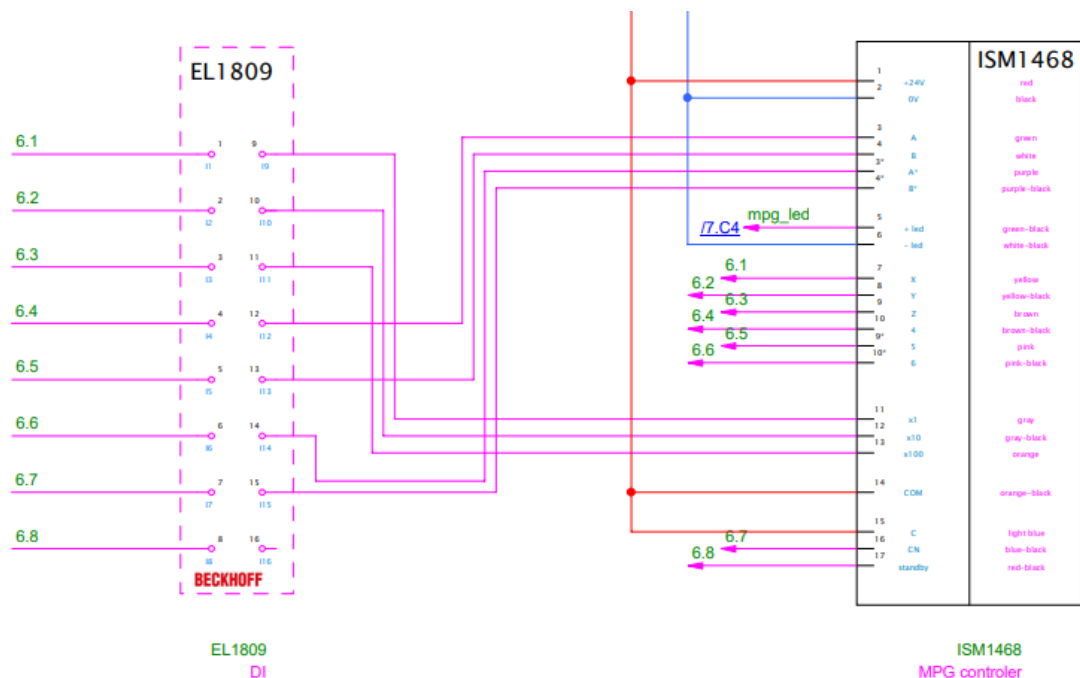
Enkodér samotný je napájen 24V, ne na obrázku uvedenými 5V.

Na svorku COM (14) se připojí společný (nulový) potenciál, k obsluze přepínačů. Jelikož se jedná o přepínače připojené do vstupní karty spínané 24V, je zde také společným potenciálem 24V.



Obr. 4.15: Značení MPG ovladače [27]

Připojení ručního ovladače je realizováno přes vstupní kartu EL1809, viz obr. 4.16. Ovládání toho ovladače je možné po zmáčknutí tlačítka na jeho boku, které doslova připojí přivedené napájení.



Obr. 4.16: Zapojení ručního ovladače (MPG)

4.3.7 Funkce a popis vnějšku hlavního rozvaděče

Obrázek č. 4.18 znázorňuje čelní stranu rozvaděče. Nachází se zde hlavní vypínač dvě ledky, zelená značí zapnuté řízení (IPC) a červená označuje stroj v chybě, a ethernetový konektor. Dále tlačítka připojené do vstupů PLC a kontrolka ovládaná z jeho výstupu.

Kontrolka ve spodní části značí aktivované motory stroje a má také funkci tlačítka, které je deaktivuje. Šest tlačítek zaštiťuje funkci programového-manuálního pohybu os X+ , X- , Y+ , Y- , Z+ , Z-. Přepínač slouží pro přepnutí mezi ovládáním z rozvaděče, pomocí zmíněných tlačítek, (poloha horizontální) a ostatním ovládáním, např. pomocí vizualizace, (poloha vertikální).

Toto ovládání na rozvaděči slouží spíše jako pohotovostní, například při výpadku ostatních řídicích prvků, jako třeba monitoru s vizualizací.



Obr. 4.18: Čelní strana hlavního rozvaděče



Obr. 4.17: Napájecí rozvaděč

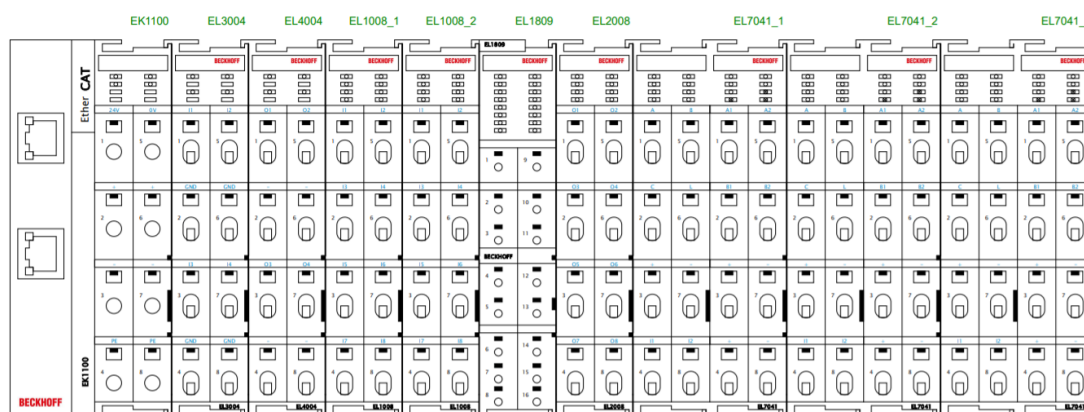
4.3.8 Vedlejší rozvaděč

Vedlejší rozvaděč o rozměrech 300x400x170 mm je použit pro montáž spínaných zdrojů, potřebných k napájení řízení krokových motorů. Dále jsou zde svorkovnice pro oddělení vnitřních a vnějších rozvodů. Náhled zapojeného rozvaděče se nachází na obr. 4.17.

Tento rozvaděč je připevněn k levé straně hlavního rozvaděče. Otvory v přiléhajících stěnách rozvaděčů je přivedeno síťové napětí a také vyvedeno napájení terminálů.

4.3.9 Náhled fyzického zapojení řízení

Pomocí programu ProfiCAD je vytvořeno elektrotechnické schéma, které obsahuje i náhled seskládaných řídicích terminálů. Nakreslené fyzické zapojení terminálů, jak by měly být následně poskládány je na obrázku 4.19.



Obr. 4.19: Elektrotechnické schéma připojení terminálů

5 TWINCAT 3, A JEHO NASTAVENÍ

Tato kapitola uvede čtenáře do problematiky softwaru TwinCAT 3 a také popisuje jeho chování. Následně je zde také nastíněno nastavení některých důležitých prvků, potřebných pro správnou funkci této aplikace. V neposlední řadě je zde odkaz na průmyslovou sběrnici.

5.1 TwinCAT 3

TwinCAT 3 (The Windows Control Automation Technology) je software tvořený společností Beckhoff pro tvorbu a vývoj automatizačních systémů na bázi PC. První řada tohoto nástroje s názvem TwinCAT byla uvedena již v roce 1986. Hlavní myšlenkou bylo zapojit do realizace automatizačních systémů PC techniku. Tento aspekt vede k jeho největší výhodě a zároveň i nevýhodě, operačnímu systému Windows, který však již není jedinou volbou.

Jedná se tedy o vývojový nástroj umožňující změnu téměř jakéhokoliv PC na systém pro řízení v reálném čase s prostředím pro běh PLC programů, tzv. runtime. Nástroj TwinCAT 3 je dalším vývojovým stupněm předchůdce TwinCAT 2, oproti němu však přináší nové možnosti při řešení komplexních úloh v oboru průmyslové automatizace, informačních technologií apod. Použitá verze TC: 3.1.4024.12.

Mezi vlastnosti zejména patří:

- vývojové prostředí TcXaeShell obdobou Microsoft Visual Studio® Shell
- možnost integrace TwinCATu do Microsoft Visual Studio®
- jeden software pro programování i konfiguraci hardwaru
- podpora objektového programování v PLC jako rozšíření standardního zápisu PLC kódu podle normy IEC 61131-3, jež obsahuje např.:
 - ST – structured text (strukturovaný text – celý kód ve formě textu)
 - LD – ladder logic diagram (žebříkový systém převzatý z reléových schémat)
 - FBD – function block diagram (program tvořen vhodným použitím logických funkčních bloků)
 - SFC – sequential function chart (použití sekvenčních funkcí do struktury vývojového diagramu)

- možnost využití C/C++ jazyka pro programování reálných aplikací
- možnost navázat na Matlab®/Simulink®
- podpora vícejádrových procesorů, s možností přidělení jader procesům a také nastavení procentuálního využití pro sdílené procesy
- možnost přenesení projektu ze softwaru TwinCAT 2
- nový způsob programování bezpečnostních funkcí
- nový modul pro efektivní monitorování: ScopeView (tzn. osciloskop)
- možnost počítačového vidění s modulem Vision
- dvě varianty Vizualizace skrz PLC HMI a TwinCAT HMI (novější)

Architekturu TwinCAT 3 tvoří dvě základní části a to vývojové prostředí eXtended Automation Engineering (XAE) a prostředí eXtended Automation Runtime (XAR).

- **Vývojové prostředí eXtended Automation Engineering (XAE)** pro TwinCAT 3 je možné nainstalovat na kterýkoliv PC minimálně s Windows 7, procesorem o frekvenci 1,6 GHz, pamětí RAM s velikostí 2 GB, grafickou kartou podporující rozhraní DirectX9 a min. rozlišení 1024x768. Dalším požadavkem je místo na disku o minimální velikosti 3 GB, nutno počítat s větší velikostí při použití dodatečných funkčních prvků.

Vývojové prostředí TwinCAT 3 obsahuje dvě základní části: hardwarovou konfiguraci k nastavení vlastností PLC, vstupů a výstupů, motion ovládání a Safety (bezpečnostní prvky) a programovací prostředí s podporou objektového programování PLC.

Vývojové prostředí již obsahuje runtime, netřeba dodatečně instalovat.

- **Runtime – eXtended Automation Runtime (XAR)** je software umožňující spouštět pouze aplikační programy v reálném čase. Lze instalovat bez vývojového prostředí XAE a je vhodné pro použití u embedded zařízení či zařízeních s nižším výkonem.

Spouští se již vytvořený program, který nadále již není možné editovat v XAR, pouze za opětovaného použití XAE. [32], [33]

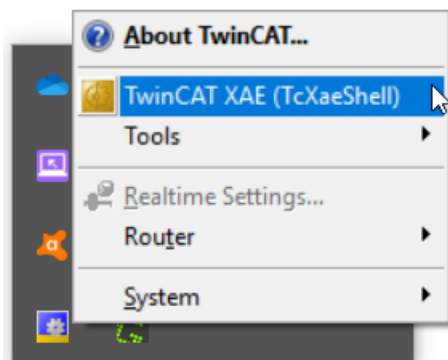
5.1.1 Vytvoření projektu

Pro vytvoření prvního projektu je nutné si stáhnout vývojové prostředí XAE TwinCATu. Software je zcela zdarma ke stažení na oficiálních stránkách společnosti Beckhoff.

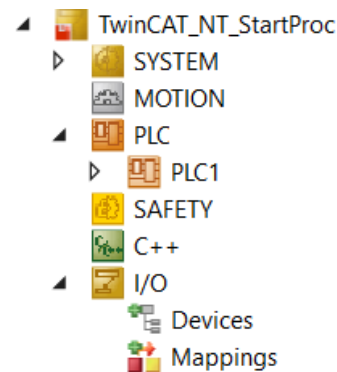
Oproti předchozí verzi TC, kde bylo nainstalováno s Visual Studio Shell, je nyní vývojové prostředí nahrazeno TcXaeShell s drobnými rozdíly vůči původnímu Studiu. Vytvoření projektu je popsáno v literatuře Retrofit/digitální řízení CNC manuální vrtačkofrézky BF20L [25] na straně 33.

TwinCAT spustíme samotnou aplikací nebo použitím ikony v tray (ikony vpravo dole u hodin), patrné z obr. 5.1.

Zde se nachází modrá ikonka „TwinCAT Config Mode“. Pravým kliknutím tuto ikonku rozbalíme a spustíme TwinCAT. V případě zelené barvy je TwinCAT v „Run Mode“, červené ve „Stop Mode“.



Obr. 5.1: Spouštění TwinCAT XAE



Obr. 5.2: Struktura TC projektu

5.2 Struktura TwinCAT projektu

Struktura TC projektu je složena z několika částí. Základní jsou „Systém“, „Motion“, „PLC“ a „I/O“, viz obr. 5.2. Dalšími použitými prvky mohou být „Safety“, „C++“, „Analytics“, „Vision“ či „CAM“.

- **Systém** – zaštiťuje nastavení celkového projektu
 - **License** – správa licencí, v případě testování lze vygenerovat každých sedm dní sedmidenní Trial licence zdarma.
 - **Real-Time** – poskytuje správu jader procesoru při použití řízení a daném projektu (Izolovaná a sdílená jádra mezi Windowsem a TC).
 - **Tasks** – nastavení úloh, jejich cyklů a priorit.
 - **Routes** – možnost připojení k jiným zařízením skrze AmsNetId.
 - **Type System a TcCOM Objects** – nastavení, pomocí kterých lze v propojení s TC přistupovat k určitým aplikacím nebo dokonce do samotného operačního systému průmyslového počítače.
- **Motion** – obsahuje parametry pro správnou funkci motorů a jejich přidružených prvků, jako například enkodéru pro zpětnovazební řízení. Propojení Motion s PLC se provádí pomocí struktury AXIS_REF.
- **PLC** – položka pro vložení PLC projektu, tzn. vytvoření celkového programového řízení stroje (celku) v závislosti na zhotovených či předpřipravených funkčních blocích, programech, strukturách, metodách a ostatních prvcích.
- **I/O** – umožňuje naskenování topologie vstupně-výstupních modulů stroje, „Hradwarovou konfiguraci“. Jednotlivé I/O terminály, jako jsou například zobrazeny v kapitole 3.3.2, obsahují vlastní nastavení a je možné k nim nalinkovat příslušné proměnné.

5.2.1 Využití licence a knihovny

- **Licence nutné pro korektní funkci programu:**
 - TC1200 – TC3 PLC
 - TF2000 – TC3 HMI Server
 - TF5000 – TC3 NC PTP
 - TF5100 – TC3 NC I
- **Funkční bloky použité v programu vycházejí z těchto knihoven:**
 - Tc2_MC2
 - Tc2_NCI
 - Tc2_PlcInterpolation
 - Tc2_Standard
 - Tc2_System
 - Tc2_Uutilities
 - Tc3_Module

5.3 Deklarace a linkování I/O proměnných

Obsluha akčních členů nebo získávání dat ze senzorů je možná v případě vytvoření určitých I/O proměnných a jejich připojení k danému terminálu.

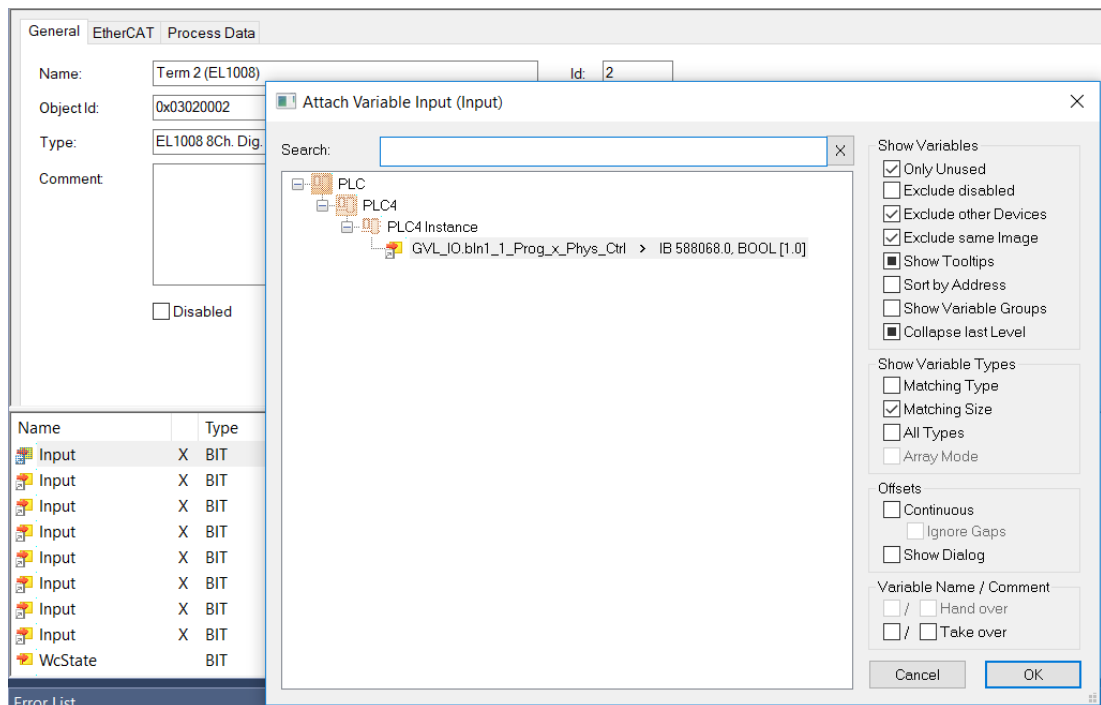
Deklarace proměnných se provede nejčastěji v „Global variable listu“ (GVL) takto:

```
<name_var> AT %<x>* :<data_type>;
```

- name_var – libovolně zvolený název proměnné
- x – I → vstupní proměnná, Q → výstupní proměnná
- data_type – datový typ proměnné (např.: bool, int, atd.)

Po deklaraci je možné proměnné nalinkovat na příslušné zařízení → terminál a kanál či bit. Např.: „Device1 → Term 2 (EL1008) → Input 4“. Podle obrázku 5.3 klikneme na daný bit a vybereme podle názvu deklarovanou proměnnou.

Pokud není možné nalézt právě zadeklamovanou proměnnou je vhodné „Buildnout“ (přeložit) projekt nebo „Aktivovat konfiguraci“, poté bude dostupná.



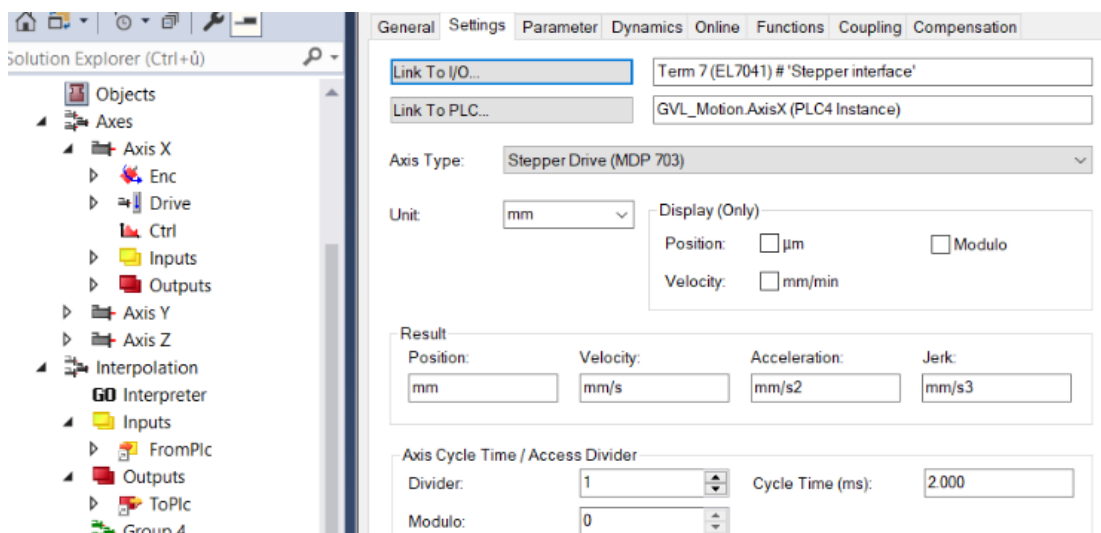
Obr. 5.3: Proces linkování proměnných

5.4 Nastavení parametrů pohonů

Nastavení správných parametrů os, popř. motorů, je jedna z nejdůležitějších základních částí projektu, na které by měl člověk myslet. V opačném případě mohou motory fungovat špatně nebo vůbec.

Použití vytvořené osy v projektu je možné za předpokladu, že je vytvořena požadovaná „AXIS_REF“ struktura pro práci s motory, která je příslušně nalinkovaná na danou osu. Stejně tak musí být nalinkován k ose příslušný motorový terminál, jak je možné vidět na obrázku 5.4 v pravé horní části.

K tomuto nastavení je možné se dostat po kliknutí na požadovanou osu a následně položku „Settings“. Na obrázku je to například: „Axis X -> Settings“. V této záložce je také možnost nastavení základních jednotek os. V tomto projektu jsou jednotkou zvoleny milimetry z důvodu odkazování se na pohyb samotné frézky v mm.



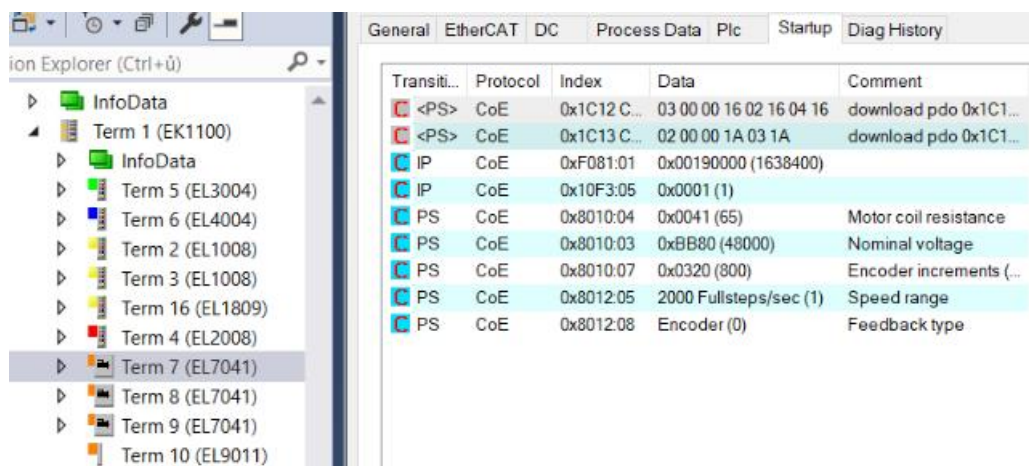
Obr. 5.4: Struktura os a jejich nastavení

Nastavení parametrů terminálu, popřípadě odvozené osy, se odvíjí od typu použitého motoru. V případě použití enkodéru je nutné zahrnout i jeho parametry. Nastavené parametry terminálů jsou detailněji popsány v kapitole 5.4.1. Hlavní nastavené parametry os jsou shrnuty v kapitole 5.4.2.

5.4.1 Nastavení parametrů terminálů

Nastavení parametrů daného terminálu se provede jeho rozkliknutím v hardwarové konfiguraci a následně pokud je stroj připojený, v čase kdy jsou měněny parametry, k vývojovému prostředí, lze tyto parametry měnit v záložce „CoE(online)“.

Další možností je přidat parametry do „Startup“ listu. Výhodou tohoto nastavení je, přehlednost použitých parametrů, jejich uchování a to i v případě výměny terminálu. Vložení např.: „Term 7 -> Startup -> New“ a zde se vloží index daného parametru a jeho požadovaná hodnota, viz obr. 5.5.



Obr. 5.5: Hardwarova konfigurace (vlevo), Startup list parametrů (vpravo)

Použité parametry vychází z typu motoru a enkodéru, jak již bylo zmíněno výše. Větší zamyšlení vyžaduje parametr „0x8010:07 (Encoder increments)“, kdy použitý enkodér čítá 200 pulzů na otáčku. Ovšem obsahuje dva signály „A a B“ a terminál EL7041 počítá jak náběžnou, tak sestupnou hranu. Tudíž má hodnotu 800. Parametry terminálu jsou shrnuty v tabulce 5.1.

Tab. 5.1: Parametry terminálu EL7041

Index	SubIndex	Název	Hodnota
0x8010	03	Nominal Voltage	48000 [mV]
0x8010	04	Motor Coil Resistance	65 [Ω]
0x8010	07	Encoder increments	800 [inc]
0x8012	05	Speed range	2000 [fullstep/s]
0x8012	08	Feedback Type	Encoder

5.4.2 Nastavení parametrů os

Nastavení některých parametrů, jako například „Maximum Velocity“, „Scaling Factor Numerator“, atd. vyžadují výpočet. Maximum Velocity (Maximální rychlost) je vypočítána z CoE parametrů tímto způsobem:

$$\text{Maximum Velocity} = \frac{\text{Speed Range}}{\text{Motor frequency}} = \frac{2000}{200} = 10 \text{ rev/s} \quad (1)$$

- Speed Range – parametr 0x8012:05 [Fullsteps/s]
- Motor frequency – frekvence použitého motoru [Hz]

Další důležitý parametr osy je Reference Velocity (Referenční rychlost):

$$\text{Reference Velocity} = \frac{\text{Maximum Velocity}}{\text{Dist.per Revolution}} = \frac{10}{1} = 10 \text{ mm/s} \quad (2)$$

- Maximum Velocity – vypočítaná maximální rychlost [mm/s]
- Dist. per Revolution – uražená vzdálenost motoru za 1 otáčku [mm/rev]

Použité nastavené parametry jsou v tabulce 5.2.

Tab. 5.2: Nastavené parametry os

Parametr	Osa			Jednotky
	X	Y	Z	
Reference Velocity	10	10	10	mm/s
Maximum Velocity	10	10	10	mm/s
Default Acceleration	55	55	55	mm/s ²
Default Deacceleration	55	55	55	mm/s ²
Homing Velocity (cam)	4	4	4	mm/s
Homing Velocity (off)	0,5	0,5	0,5	mm/s
Manual Velocity (Fast)	8	8	8	mm/s
Manual Velocity (Slow)	2	2	2	mm/s
Rapid Traverse Velocity	5	5	5	mm/s
Kv Factor (standstill)	4	4	4	mm/s/mm
Kv Factor (moving)	2	2	2	mm/s/mm

Nastavení parametrů enkodéru v záložce „Enc“ se týká především „Scaling Factor Numerator (SFN)“ a „Scaling Factor Denominator (SFD)“, v případě použití externího enkodéru se vypočítají následujícím způsobem:

$$\text{Scaling Factor} = \frac{\text{Dist.per Revolution}}{(\text{Increments} \cdot 4)} = \frac{1}{(200 \cdot 4)} = 0,00125 \text{ mm/INC} \quad (3)$$

- Dist. per Revolution – uražená vzdálenost motoru za 1 otáčku [mm/rev]
- Increments – počet pulzů enkodéru [inc/rev]

V nastavení lze zapsat jak desetinným číslem: SFN = 0,00125 a SFD = 1 nebo použitím zlomku: SFN = 1 a SFD = 800.

Parametr „Dist. per Revolution“ je uražená vzdálenost za jednu otáčku, vychází tedy z tabulky 2.1, kde jsou uvedeny stoupání trapézových šroubů jednotlivých os a také z převodového poměru použitého převodu s ozubenými koly, který je roven 1:2. Pro znázornění je vypočten tímto způsobem:

$$\text{Dist. per Rev.} = p \cdot i = 2 \cdot \frac{1}{2} = 1 \text{ mm} \quad (4)$$

- p – stoupání trapézového šroubu osy
- i – převodový poměr [mm]

Pro úplnost jsou v tabulce č. 5.3 doplněny maximální dosažitelné vzdálenosti jednotlivých os.

Tab. 5.3: Nastavené parametry „Enc“

Parametr	Osa			Jednotky
	X	Y	Z	
Soupání trapézového šroub.	2	2	4	mm
Scaling Factor Numerator	1	1	2	mm/ot
Scaling Factor Denominator	800	800	800	INC/ot
Minimum Position	0	0	-245	mm
Maximum Position	420	125	0	mm

5.5 Boot projekt

Součástí každého projektu je také nastavení jeho spuštění po zapnutí počítače. TwinCAT tuto problematiku řeší několika kroky.

- 1) Spuštění „Runtime mode“ po startu IPC (běžně se spouští „Config mode“), dalo by se říct nastartování virtuálního PLC se provede ve složce „SYSTEM -> Settings“, kde se zaškrtně položka „Run Mode (Enable)“. Tuto položku musíme potvrdit pomocí tlačítka „Apply“.
- 2) Dále se pokračuje nastavením boot projektu, provedeme pravým kliknutím na náš PLC projekt, kde zvolíme „Activate Boot Project..“ a následně zaškrtneme položku „Autostart Boot Project“.
- 3) Po provedení těchto kroků je nutné aktivovat konfiguraci projektu. Následně po startu IPC již vše běží a je možné ovládat příslušnou vizualizaci.

5.6 Nastavení HMI serveru

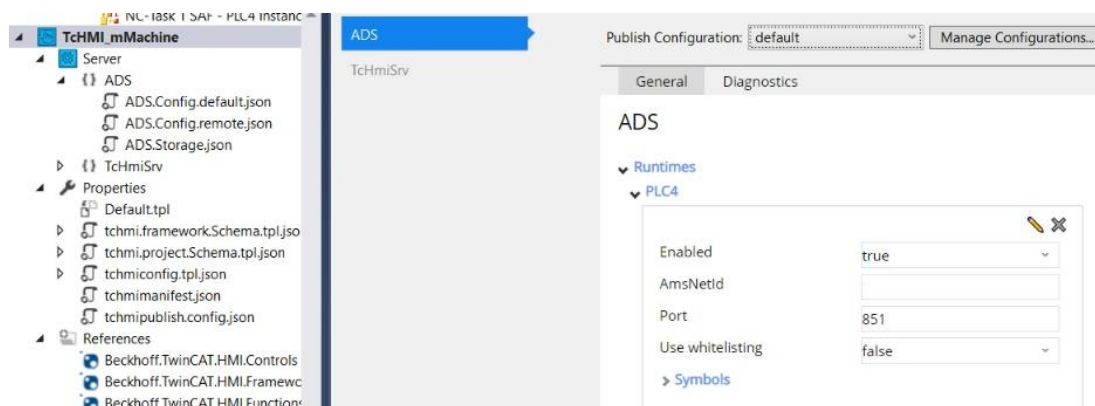
Spuštění vizualizace bez vývojového prostředí je podmíněno instalací HMI serveru. Toto je umožněno stažením a instalací funkce TF2000 z oficiálních stránek společnosti Beckhoff. Použitá verze HMI je 1.12.746.0.

Po instalaci se objeví modrá ikonka serveru v tray, kterou rozklikneme pravým tlačítkem a zvolíme položku „Config“. Při prvním spuštění se objeví přihlašovací stránka, kde je nutné si zvolit přihlašovací údaje. Dalším spuštěním se zapne prohlížeč s příslušnou stránkou pro konfiguraci. Reset serveru je z důvodu bezpečnosti umožněn pouze při reinstalaci.

Celá vizualizace je na pozadí prováděna v JavaScriptu (JS) a na její spuštění je nutné používat prohlížeč s HTML5. V konfiguraci je vhodné v záložce „TcHmiSrv -> WebServer“ přidat „Endpoint“ v důsledku toho, že budeme používat http protokol. Tudiž vložíme `http://0.0.0.0:1010`, aby bylo možné nahrát projekt na server z externího vývojového prostředí.

Nastavení propojení Runtimu (PLC) a HMI serveru se provede ve stromové struktuře HMI projektu. Přesně v záložce „Server -> ADS -> General“ pod pojmem „Runtimes“.

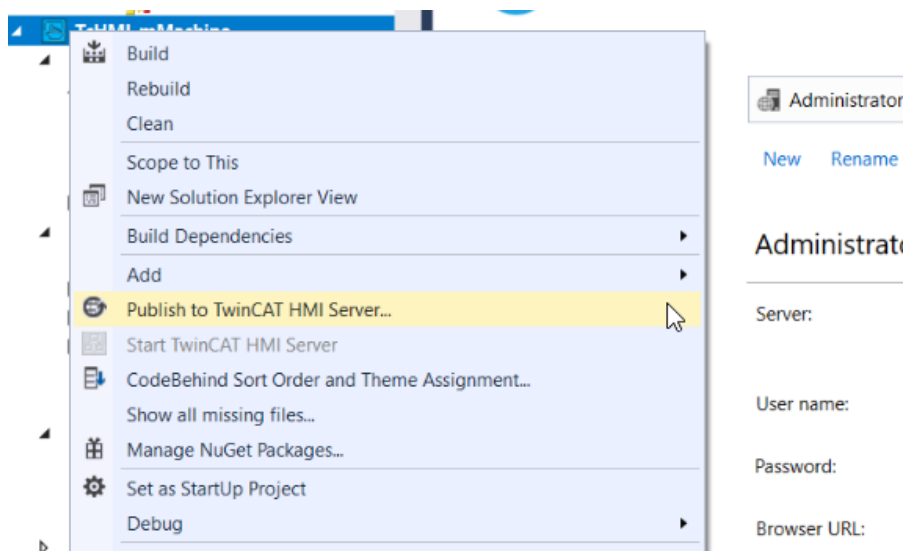
Jak je patrné na obrázku 5.6, zde je potřeba přidat runtime s názvem shodným s názvem PLC projektu, v našem případě „PLC4“. Poté je nutné nastavit AmsNetId TwinCATu, kde projekt běží a port runtime, běžně „851“ pokud není používáno více PLC projektů.



Obr. 5.6: Propojení PLC a HMI serveru

Následně pokud PLC projekt běží, je možné linkovat proměnné do HMI.

Nahrání projektu na server neboli „Publish to TwinCAT HMI Server“ je ukázáno na obr. 5.7. Zde nutné zadat IP adresu a port serveru, např.: „1.1.1.1:1010“ a příslušné přihlašovací údaje vyplněné po instalaci HMI serveru. [34]



Obr. 5.7: Publish to TwinCAT HMI Server

Jelikož HMI server a TwinCAT (PLC) nabíhají po startu PC stejně, HMI server často rychleji. Je vhodné opozdit spuštění HMI server z důvodu nechtěných error stavů, které by mohly být detekovány v případě, nezinicializovaných hodnot z PLC.

Řešením je například vytvoření „.bat“ souboru. V tomto souboru je důležitý příkaz „TIMEOUT x“, kde x je zadaný čas v sekundách, a také cesta k aplikaci HMI serveru.

Funkčnost bat souboru po staru PC je poté zajištěna vložením jeho zástupce, ne samotného souboru, do „StartUp“ složky Windowsu.

5.7 EtherCAT a výběr sběrnic

Volba průmyslové sběrnice je většinou považována za druhořadou záležitost a volí se spíše řídicí systém, který má již přiřazenou vlastní sběrnici. Vezmeme-li v úvahu tu menší skupinu lidí, kteří opravdu vybírají sběrnici v závislosti na výkonu a dalších parametrech, z jakého důvodu skončí jejich výběr právě na EtherCATu?

EtherCAT je technologie průmyslového Ethernetu v reálném čase, která pracuje v takzvaném režimu „on the fly“, tedy zpracování informací za chodu. Přesněji to znamená, že EtherCAT Master vyšle EtherCATový frame (rámec), který postupně prochází topologií zapojené sběrnice, při průchodu obsluhuje daná zařízení a je zpožděn pouze o čas průchodu hardwarem. Při posledním uzlu se detekuje otevřený port a zpráva se odešle zpět Masteru. Tato vlastnost umožňuje využít přes 90% datového toku a použít krátké časové cykly (menších než 100 μ s) s nízkou odezvou pro přenosovou synchronizaci (menší než 1 μ s).

Další velkou výhodou EtherCATu je možnost využití téměř jakékoliv topologie, kde navíc odpadá nutnost použití switchů (umožňuje obsluhu prvků v závislosti na jejich adresách) a s tím ruku v ruce jsou na sběrnici požadovány i nižší náklady. EtherCAT také umožňuje již integrovanou bezpečnost. Adresace prvků v síti je řízena automaticky Masterem, což vede k jednodušší instalaci sběrnice. EtherCAT je hojně využíván společnostmi ze skupiny „EtherCAT Group“ (Beckhoff, Samsung, Omron, ...), které obohacují portfolio výrobků na této sběrnici. [35], [36]

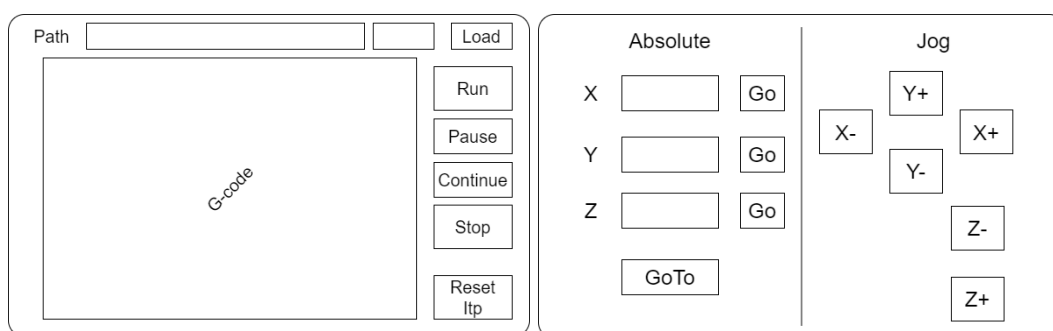
V pravém horním rohu téhož obrázku jsou dvě kontrolky. Run značí běh zařízení a Error jeho poruchu. Uprostřed obrazovky se nachází dvě okna, jejich funkce závisí na výběru okna ze seznamu nad daným oknem. Více informací je popsáno níže.

Spodní část obrazovky obsahuje ovládání společné pro různé typy řízení nebo také prvky, u kterých je nutné stálé zobrazení. Popis jednotlivých sekcí:

- **Tool** – nastavení korekce nástroje.
- **Coordinates shift** – posun či výběr různých souřadnicových systémů. Použitelné pro určení počátku obrobku.
- **Motors** – ovládání motorů. Reset motorů, možnost zreferování os a dále také posun na domovskou pozici.
- **Spindle** – ovládání vřetena. Nastavení a zobrazení jeho aktuální rychlosti.
- **Coordinates** – zobrazení souřadnic v milimetrech, s možností sledování aktuálních rychlostí jednotlivých os.
- **Feed** – nastavení rychlosti posuvu.

Popis levé střední sekce:

- **Machine** – zobrazení modelu stroje.
- **Graph** – vykreslení pohybu nástroje.
- **Info_parameters** – tabulka dodatečných informačních parametrů stroje.
- **EventLogger** – tabulka událostí, v závislosti na jejich nastavení.



Obr. 6.2: Návrh automatického řízení (vlevo), manuálního (vpravo)

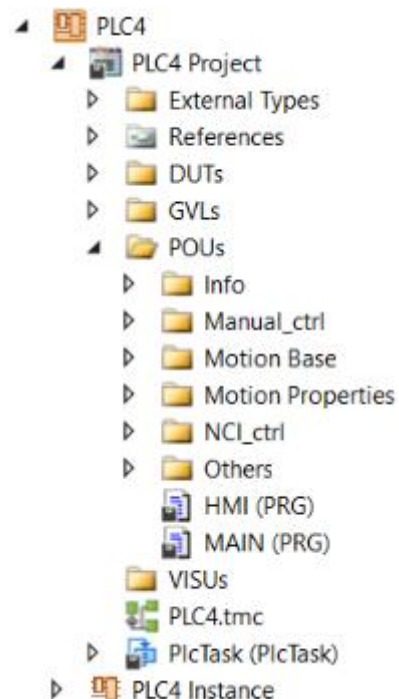
Popis pravé střední sekce:

- **Auto** – automatické řízení, viz obrázek 6.2 (vlevo).
 - **Path** – určuje cestu ke zdrojovému souboru.
 - **Load** – načte zdrojový soubor.
 - **G-code** – textové pole pro zobrazení aktuálně načteného souboru.
 - **Run** – spuštění automatického obrábění.
 - **Pause** – pozastavení automatického obrábění.
 - **Continue** – pokračování v pozastaveném programu.
 - **Stop** – zastavení stroje.
 - **Reset Interpreter** – reset aktuální skupiny motorů, která je ovládána automatickým řízením.
- **Manual** – programové manuální ovládání, viz obrázek 6.2 (vpravo).
 - **Absolute** – Absolutní zadávání souřadnic, na které se má nástroj následně posunout. Posuv každé osy zvlášť nebo dohromady.
 - **Jog** – kontinuální manuální pohyb ovládaný pomocí tlačítek. Pohyb započne v případě zmáčknutí tlačítka a skončí při jeho puštění.
- **Manual_physic** – ruční ovládání pohybu frézky bez programového ovládání, tzn. motory jsou vypnuty.

6.2 Struktura programového řízení

Struktura programu se snaží brát zřetel na základní používané principy při návrhu PLC programů a zároveň apeluje na jednoduchost a schopnost orientace v základní stromové struktuře PLC projektu.

V závislosti na těchto programech jsou vytvořeny příslušné struktury s určitými proměnnými, jež zajišťují přiřazení daných proměnných k daným programům. Tyto struktury také zpřehledňují kód, snižují náročnost při změnách programu a umožňují vícenásobné použití v případě jejich deklarace.



Obr. 6.3: Struktura PLC projektu

Vytvoření instancí těchto struktur je provedeno v „Global Variable Listu“, konkrétně v „GVL_Variables“. Další GVL jsou použity pro deklaraci struktur k motorům a také pro deklaraci I/O proměnných. Konkrétně jsou to „GVL_Motion“ a „GVL_IO“.

Struktura PLC je vyobrazena na obrázku 6.3. Vytvořené složky pod složkou „POUs“ obsahují vytvořené programy, funkční bloky a metody vztahující se k určitým prvkům řízení. Jednotlivé sekce mají funkce:

- **Info** – tato složka obsahuje programy týkající se výčtu a zobrazení doplňujících informací o stroji a elektronice.
- **Manual_ctrl** – obsahuje programy v souvislosti s manuálním ovládáním, jako např. samotné manuální ovládání či ovládání ručního ovladače.
- **Motion Base** – obsahuje funkční bloky a metody vztahující se přímo k danému motoru, je zde např. blok na zapnutí motoru či jeho reset.
- **Motion Properties** – obsahuje programy doprovodné k samotnému pohybu os. Jde tedy o programy odrážející spodní sekci z obrázku č. 6.1.
- **NCI_ctrl** – nachází se zde programové vybavení potřebné k automatickému řízení. Např. metoda se stavovým automatem.
- **Others** – tato složka uchovává nezařazené programy.

Dále jsou zde hlavní programy:

- **MAIN** – hlavní část PLC programu. Tento program je propojen s taskem a je tedy cyklicky volán. V tomto hlavním programu jsou volány všechny ostatní programy, u kterých požadujeme jejich cyklickou obsluhu. Jsou zde také uvedeny podmínky pro výběr typu řízení.
- **HMI** – Program určený k předání povelů, stavů a parametrů mezi PLC a HMI.

6.2.1 Výměna proměnných s HMI

Struktury použité v programech mají většinou název shodný s názvem programu a jsou tudíž vytvořeny pro použití k jedné dané problematice, která je v témže programu řešena.

Při návrh struktur napojených do HMI je poté brán zřetel na možné standardy, používané při psaní PLC kódu a s ním spojených programů.

HMI struktury jsou rozděleny na tyto části:

- **Cmd** – příkazy (commands)
 - jedná se o fyzicky namapované vstupy či vstupní proměnné propojené s vizualizací, které vytvářejí interakci s okolím
 - po provedení příkazu, by měl být příslušný příkaz vynulován
 - např. zapni motor
- **States** – stavy
 - jedná se o fyzicky namapované výstupy nebo výstupní proměnné, které určují, v jakém stavu se daný modul nachází
 - např. motor je zapnut
- **Param** – parametry (parameters)
 - mohou mít statickou hodnotu nebo svojí hodnotu měnit v závislosti na určitém příkazu
 - např. aktuální rychlost motoru

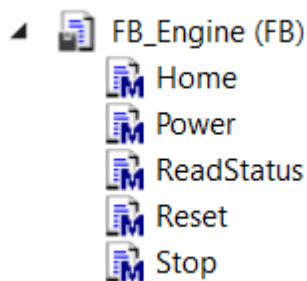
Jak již bylo zmíněno výše, výměnu informací mezi těmito dvěma skupinami zajišťuje program „HMI“.

6.3 Základní funkční bloky pohybů

Složka „Motion Base“ obsahuje dva funkční bloky. První se vztahuje k motorům, „FB_Engine“, kdy tento blok slouží jako šablona pro použití motoru a jeho základních funkcí. Druhý blok, „FB_Move_Type“ zajišťuje základní pohyby.

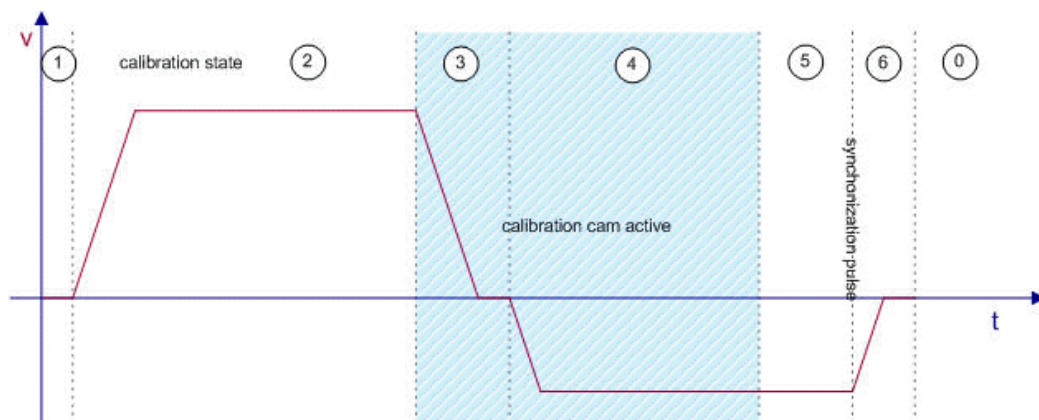
6.3.1 Funkční blok motorů

Blok po své inicializaci vytvoří referenci na odkazovanou osu a tuto referenci ověří příkazem „__ISVALIDREF()“. Následně zjistí stav osy. Ve funkčním bloku je vytvořeno pět metod (funkcí). Z obrázku 6.4 je patrné, že tyto metody jsou: „Power“, „ReadStatus“, „Reset“, „Stop“ a „Home“.



Obr. 6.4: FB_Engine

- **Power** – metoda se odkazuje na blok „MC_Power“ a hlavní funkcí je zajištění aktivace motoru. Hlavní parametry jsou:
 - **Axis** – reference na používanou osu.
 - **Enable** – aktivace, reakce na stav „true“.
 - **Enable_Positive** – aktivace směru po směru hod. r., reakce na stav.
 - **Enable_Negative** – -“- proti směru hod. r., reakce na stav.
- **ReadStatus** – zjištění stavu osy, blok „MC_ReadStatus“.
 - **Axis** – reference na používanou osu.
 - **Enable** – aktivace, při „TRUE“ se stav aktualizuje každým voláním.
- **Reset** - reset motoru blokem „MC_Reset“, popřípadě zjištění id chyby.
 - **Axis** – reference na používanou osu.
 - **Enable** – aktivace, reaguje na náběžnou hranu.
- **Stop** – zastavení osy, blok „MC_Stop“. Po aktivaci musí být použit „Reset“.
 - **Axis** – reference na používanou osu.
 - **Execute** – aktivace, reaguje na náběžnou hranu.
- **Home** – metoda používá blok „MC_Home“. Tato funkce referuje osy předem nastaveným způsobem. Nastaveno je referování na koncové čidlo.
 - **Axis** – reference na používanou osu.
 - **Execute** – aktivace, reaguje na náběžnou hranu.
 - **Position** – volba souřadnic, které se nastaví po dokončení referování.
 - **HomingMode** – „MC_DefaultHoming“, spustí standardní referování.
 - **bCalibrationCam** – signál čidla, na které je referování prováděno.



Obr. 6.5: Stavy referování [37]

Z obrázku 6.5 je patrné, že referování probíhá ve dvou krocích. Prvním krokem je najetí k čidlu referenční rychlostí. Poté se posun zastaví a následuje druhý krok, kdy se otočí směr pohybu a motor pomalu sjíždí z čidla. Jakmile se přestane detekovat hodnota na čidlu, nastaví se pozice a referování je u konce.

6.3.2 Funkční blok pohybů

Metody toho bloku tvoří základní pohyby, jako jsou: „Jog“ – kontinuální pohyb, „MoveAbsolute“ – pohyb na předem definované souřadnice a „MoveRelative“ – posun o určitou vzdálenost, tento posun je například použit u MGP ovladače.

MC_Jog

Funkce Jog zastupuje kontinuální pohyb. Spustí se náběžnou hranou a ukončí stavem v nule. Funkce obsahuje pár základních argumentů:

```
fbAxisJog(Axis:= refAxis, JogForward:= bAxisJogFwd, JogBackwards:=  
bAxisJogBwd, Mode:= MC_JOGMODE_CONTINUOUS, Position:= , Velocity:=  
fAxisVelocity, Acceleration:= , Deceleration:= , Jerk:= , Done=> , Busy=>  
bJogBusy, Active=> , CommandAborted=> , Error=> , ErrorID=> );
```

- **Axis** – reference na používanou osu.
- **JogForward** – aktivace na náběžnou hranu, směr dopředu.
- **JogBackward** – aktivace na náběžnou hranu, směr dozadu.
- **Mode** – MC_JOGMODE_CONTINUOUS – aktivní dokud „true“.
- **Velocity** – Nastavená rychlost posuvu osy.
- **Busy** -> detekce, zda byl pohyb dokončen a může být vykonán další.

MC_MoveAbsolute

Posun na předem určené souřadnice. Funkce obsahuje pár základních argumentů, argumenty popsány výše již budou vynechány:

```
fbAxisMoveAbs(Axis:= refAxis, Execute:= bAxisMoveAbsExecute, Position:=  
fAxisPosition, Velocity:= fAxisVelocity, Acceleration:= , Deceleration:= ,  
Jerk:= , BufferMode:= , Options:= , Done=> bMoveAbsDone, Busy=> bMoveAbsBusy,  
Active=> , CommandAborted=> , Error=> , ErrorID=> );
```

- **Execute** – aktivace pohybu na náběžnou hranu.
- **Position** – Absolutní hodnota cílové pozice.
- **Done** -> detekce dokončení právě probíhaného posuvu.

MC_MoveRelative

Relativní pohyb, tzn. pohyb o určitou zadanou vzdálenost. Tato funkce umožňuje nastavit již kromě zmíněných argumentů ještě tyto:


```
fbAxisMoveRel(Axis:= refAxis, Execute:= bAxisMoveRel, Distance:=
fAxisDistance, Velocity:= fAxisVelocity, Acceleration:= , Deceleration:= ,
Jerk:= , BufferMode:= MC_BlendingPrevious, Options:= , Done=> , Busy=>
bMoveRelBusy, Active=> , CommandAborted=> , Error=> , ErrorID=> );
```

- **Distance** – nastavení relativní vzdálenosti, která má být uražena.
- **BufferMode** – MC_BlendingPrevious – je mód, který při obdržení dalšího potvrzení pohybu naváže na pohyb minulý s novými parametry. Pohyb nemusí být pozastaven.

6.4 Manuální řízení

Manuální řízení v sobě obsahuje program na ovládání frézky z rozvaděče, tedy „PRG_Local_Man_Ctrl“, jehož funkce je popsána v kapitole 7.1. Dále obsahuje program pro manuální řízení z vizualizace „PRG_Manual_Ctrl“ a také program „PRG_WithOut_Man_Ctrl“, který de facto deaktivuje motory. V neposlední řadě je zde program „PRG_MPG“, jež zajišťuje funkci ručního ovladače.

6.4.1 Manuální řízení z vizualizace

Základem programu „PRG_Manual_Ctrl“ je vytvoření referencí na pohybové osy, které se následně přiřadí k vytvořeným instancím funkčních bloků z 6.3.1 a 6.3.2. Instance jsou vytvořeny jako pole o třech prvcích, počet prvků roven počtu os.

Dále je zde použita metoda na aktivaci motorů, tedy „FB_Engine[i].Power();“. Písmeno i je použito v rámci cyklu „FOR“ jako obslužení všech instancí.

Program také používá metody z 6.3.2 pro kontinuální a absolutní pohyb. Jednotlivé pohyby jsou funkční v závislosti na podmínkách. Tyto podmínky zabraňují použití více druhů pohybů najednou a zajišťují tak vyvarování se chybným stavům.

6.4.2 Manuálním řízení ručním ovladačem

Řízení ručního ovladače je ze základu stejné jako manuální řízení z předchozí kapitoly. Rozdíl je v použití metody relativního pohybu, popsána v kapitole 6.3.2.

Základní jednotkou posuvu je zvoleno „0,1 mm“, odpovídající hodnotě na ovladači x1. Přičítání či odčítání pulzů, čítání vzdálenosti pro pohyb, je detekováno pomocí log. stavu určitého signálu a náběžnou hranou jiného signálu.

V závislosti jejich sledu je následně hodnota inkrementována či dekrementována. Náběžná hrana je detekována funkcí „R_TRIG“, jejíž výstup „Q“ je nastaven do jedničky v případě náběžné hrany na vstupu.

Čítání pulzů je doplněno časovačem nastaveným na 40ms, který se resetuje změnou načítané hodnoty. Pokud bude hodnota po nastavený čas nezměněna, dojde následně k samotnému relativnímu posuvu.

6.5 Automatické řízení

Automatické řízení obsažené ve složce „NCI_ctrl“ se skládá z programu „PRG_Auto_Ctrl“, kompletně vytvořeného funkčního bloku „FB_Auto_Ctrl“ a metody „StateMachine“ odkazující se k tomuto bloku.

Program obsahuje instance os a k nim přiřazené reference stejně jako v kapitole 6.4.1. Dále je zde vytvořena instance funkčního bloku „FB_Auto_Ctrl“, která je volána následujícím způsobem a obsahuje tyto argumenty:

```
_fbAutoControl(bStart:= , bPause:= , bContinue:= , bStop:= , bResetItp:=  
, sPath:= sPath, fNciShiftX:= , fNciShiftY:= , fNciShiftZ:= , nCS_State:= , bBusy=>  
, bError=> , bDone=> , AxisX:= , AxisY:= , AxisZ:= , stItpToPlc:= ,  
stPlcToItp:= );
```

- **bStart** – spuštění automatického řízení.
- **bPause** – pozastavení automatického řízení.
- **bContinue** – pokračování po pozastavení.
- **bStop** – zastavení automatického řízení.
- **bResetItp** – reset interpreteru, tzn. os podílejících se na auto. řízení.
- **sPath** – cesta ke zdrojovému „.nc“ souboru i s jeho názvem.
- **sNciShiftX** – posun souřadnic v ose X.
- **sNciShiftY** – posun souřadnic v ose Y.
- **sNciShiftZ** – posun souřadnic v ose Z.
- **nCS_State** – zvolení souřadného systému.
- **bBusy** -> detekce, zda je řízení stále zaneprázdněno (vykonáváno).
- **bError** -> detekce chybového stavu.
- **bDone** -> detekce, zda bylo právě prováděné automatické řízení dokončeno a může být vykonán další.
- **AxisX** – reference na osu X.
- **AxisY** – reference na osu Y.
- **AxisZ** – reference na osu Z.

- **stItpToPlc** – reference na speciální strukturu, používanou pro předávání informací z interpolační skupiny do PLC programu „NCTOPLC_NCICHANNEL_REF“.
- **stPlcToItp** – reference na speciální strukturu, používanou pro předávání informací z PLC programu do interpolační skupiny „PLCTONC_NCICHANNEL_REF“.

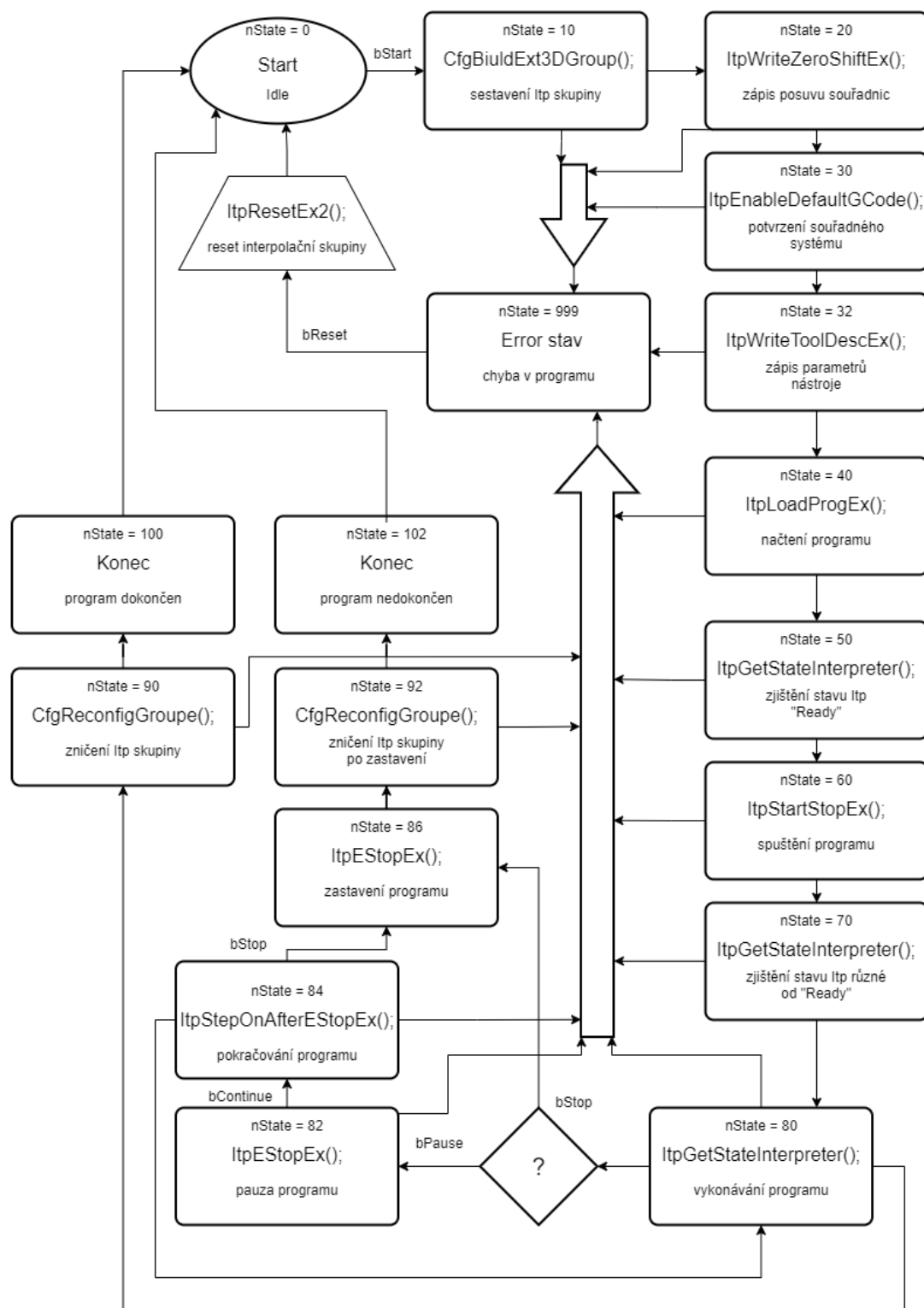
Funkční blok „FB_Auto_Ctrl“ deklaruje proměnné pro metodu „StateMachine“. Před zavoláním této metody je nutné nejprve ověřit reference os a interpolačního kanálu. Pokud reference nejsou v pořádku, nelze metodu volat. Metoda se stavovým automatem je popsána v kapitole 6.5.1.

V bloku je taky použita funkce „ItpSetOverridePercet();“ pro nastavení „overridu“ (výkonu, 0-100%) kanálu. Dále je zde prováděn odchyt M-funkcí příkazem „stItpToPlc.HskMFuncReq“, který pokud je v jedničce, zachytil požadavek na M-funkci. Číslo M-funkce já následně vyčteno „stItpToPlc.HskMFuncNo“, od něhož se poté odráží vykonání příkazů z CASE podmínky. Pokud byla definovaná funkce obsloužena správně, potvrdí své vykonání funkcí „ItpConfirmHsk()“ a vykonávání g-kódu může pokračovat.

6.5.1 Stavový automat

Stavový automat použitý k automatickému řízení je na obrázku 6.6. Výjimku zařazení do stavového automatu má v této metodě „StateMachine“ funkce „Resetu interpolační skupiny“, která může být volána v jakékoliv probíhající části stavového automatu. Z obrázku 6.6 můžeme vidět rozdělení funkcí do stavů. Při používání interpolačních funkcí je nutné se odkazovat na příslušnou referenci „stItpToPlc“. Jednotlivé použité stavy a funkce s jejich argumenty mají následující význam:

- **0:** Stav nečinnosti, čeká se na spuštění signálem start.
- **10:** Funkce sestavení interpolační skupiny.
 - nGroupeld – Id interpolační skupiny, použití „ItpGetGroupId()“;
- **20:** Přepis aktuálně použitých posuvů souřadnic.
 - sZeroShiftDesc – použití struktury „stNciShift“ se souřadnicemi
- **30:** Potvrzení požívaného souřadného systému.
- **32:** Zápis do tabulky nástrojů právě používaných korekcí.
- **40:** Načtení „nc“ souboru s příslušným g-kódem.
 - sPrg – celková cesta k souboru i s jeho názvem.
 - nLength – délka názvu, použití příkazu „LEN()“;



Obr. 6.6: Stavový automat automatického řízení

- **50:** Kontrola, zda-li je interpreter ve stavu „Ready“.
- **60:** Spuštění vykonávání programu.
- **70:** Kontrola stavu, zda-li již není „Ready“. Používá se z důvodu, aby nedošlo k předčasnému ukončení vykonávání programu.
- **80:** Vykonávání programu, čekání na dokončení.
- **82:** Pozastaveno vykonávání programu. Signál „bPause“.
- **84:** Pokračování programu po pozastavení. Signál „bContinue“.
- **86:** Zastavení chodu programu. Signál „bStop“.
- **90:** Zničení interpolační skupiny po dokončení programu.
- **92:** Zničení interpolační skupiny po zastavení programu.
- **100:** Program dokončen. Nastaví se bit „Done“ do log. 1.
- **102:** Program nedokončen.
- **999:** Error stav. Do tohoto stavu je možné dostat se z jakéhokoliv jiného stavu. Při přechodu z jednoho stavu do druhého je vždy testováno dokončení a možná chyba dané funkce. V případě chyby stavový automat přechází do tohoto stavu a je nutné použít „ResetItp“.

6.5.2 JavaScript funkce

Zobrazování aktuálního času, vyčítání používaného kódu ze souboru nebo jeho následné ukládání jsou například funkce obtížně realizovatelné v PLC programu. Jelikož je HMI ze základu programováno v JavaScriptu (JS), byla zvolena cesta vytvoření JS funkcí pro tyto účely. Následující kód zobrazuje funkci „openFileDialog“ na vyčítání textu daného zdrojového souboru.

```
function openFileDialog(dataInternalVar) {
    // create element input with type=file
    const fileSelector = document.createElement('input');
    fileSelector.type = "file";
    // register function for returned data
    fileSelector.addEventListener('change', (event) => {
        var data = event.target;
        if (!data.files) {
            alert("This browser doesn't seem to support the
`files` property of file inputs.");
        }
        else if (!data.files[0]) {
            alert("Please select a file before clicking 'Load'");
        }
        else {
            var file = data.files[0];
            var fr = new FileReader();
            TcHmi.Symbol.write("Auto_gCodeFileName",
TcHmi.SymbolType.Internal, file.name);
        }
    });
}
```

```

        // onload is called, when the FileReader read all data
        fr.onload = function () {
            // write data to HMI Internal variable
            TcHmi.Symbol.write(dataInternalVar,
TcHmi.SymbolType.Internal, fr.result, function (data) {
                if (data.error === TcHmi.Errors.NONE) {
                    // Handle success...
                } else {
                    // Handle error...
                }
            });
        });
        fr.readAsBinaryString(file);
    }
}); // open dialog for file open
fileSelector.click();
}

```

Vytvoření vstupního („input“) elementu typu „file“ umožňuje na událost „click“ otevřít dialogové okno, kde je možné vybrat zdrojový soubor. Následně se zkontroluje typ souboru, uloží jeho název a vyčte se jeho obsah jako „string“.

Při práci s těmito funkcemi je důležité mít na paměti, že se nejedná o webové stránky a přístup k určitým atributům je tudíž odlišný, jako například při práci se symboly je nutné vytvořit si interní proměnou v HMI, ke které se následně přistupuje jako:

„TcHmi.Symbol.<akce>(<název prom.>, TcHmi.SymbolType.Internal, <atribut>);“

Podobná je funkce „Download“ vytvořena pro ukládání „.nc“ programu. Kód je znázorněn níže. Důležitou funkcí bylo nahrazení znaků odřádkování „\n“ za znaky „\r\n“, tedy CR, LF, které jsou používány v souborech na odřádkování ve Windows. Dále bylo potřeba zvolit typ dat a nastavit příslušné atributy.

```

function Download(data, filename) {
    var element = document.createElement('a');
    var text = data.replace(/\n/g, "\r\n");
    element.setAttribute('href', 'data:text/plain;charset=utf-8,'
+ text);

    element.setAttribute('download', filename);

    element.style.display = 'none';
    document.body.appendChild(element);

    element.click();

    document.body.removeChild(element)
}

```

K ukládání souboru se následně používá nastavení v používaném prohlížeči.

6.6 Parametry a dodatečné ovládání

Tato kapitola směřuje na programy obsažené ve složce „Motion Properties“ a odkazuje se k zobrazeným a nastavitelným parametrům a ovládání nacházející se trvale zobrazeny ve spodní části vizualizace. Mezi tyto programy patří „PRG_ActPosition“ pro zobrazování parametrů pozic, volba souřadného systému „PRG_CoorSystem“, informace k motorům PRG_Engines“, nastavení výkonu „PRG_Override“, „PRG_Spindle“ pro ovládání vřetene a za delší zmínku stojí nastavení nástrojů v „PRG_Tool“.

- **PRG_ActPosition**

Program vyčítá aktuální souřadnice z os a zapisuje je do vizualizace. Dále je zde proveden přepočet posunutých souřadnic v případě použití odlišného souřadnicového systému než toho základního. Program také vyčítá aktuální rychlosti os. Přetypování veličin na datový typ „string“ je zde provedeno funkcí „LREAL_TO_FMTSTR();“.

- **PRG_CoorSystem**

Program na vyhodnocení nastaveného souřadného systému. Po obdržení příkazu na daný systém se nastaví stav daného systému a příkaz je vynulován. Dále jsou podle stavů používány nastavené posuvy jednotlivých os. Nastavení posuvů realizováno ve vizualizaci. Tlačítko „Set“ obsahuje událost „.onPressed“, ve které jsou souřadnice připisovány.

- **PRG_Engines**

Tento program kontroluje stav motorů. Je zde volána metoda „Reset“ funkčního bloku „FB_Engine“ pro všechny tři motory. Metoda „Home“ je také volána v tomto programu. Jako poslední je zde realizován posuv na domovské souřadnice, které jsou zvoleny jako: $[X, Y, Z] = [0, 0, 0]$.

- **PRG_Override**

Reguluje rychlost os v závislosti na lineárním posuvníku použitým ve vizualizaci. Přepis rychlostí je přepočítáván v rozmezí 0-100%, kde hodnota 100 je ekvivalentní maximální rychlosti zapsané v „Tab. 5.2: Nastavené parametry os“.

- **PRG_Spindle**

Napojení na požadované fyzické periferie je ve fázi procesu. Program je však nachystán na napojení k vřetenu. Jsou zde dva výstupy, které mají simulovat zapnutí vřetene ve a proti směru hodinových ručiček.

Ovládání otáček vřetene, tak i jejich sledování, je realizováno napěťovou smyčkou připojenou do analogových terminálů. Pro vstupní hodnoty je provedena jejich standardizace, pro výstupní destandardizace. Jelikož jsou pořízeny 12-ti bitové terminály, je jejich přepočet počítán následující způsobem s následující konstantou.

Příklad pro standardizaci: (nastavování otáček v rozmezí 0 – 100%)

„<vstupní hodnota do PLC> = (100 * <vstupní hodnota do terminálu>) / 4096;“

6.6.1 Nástrojová korekce

Správná funkce automatického režimu je podmíněna nástrojovou korekcí. Tato korekce je spravena v programu „PRG_Tool“. Výběr nástroje ve vizualizaci je realizován nástrojem „ComboBox“, který je před svým užitím nutno inicializovat. Inicializace je provedena strukturou o třech prvcích: „id, text a value“. Instance této struktury je inicializována takovým počtem, kolik požadujeme parametrů (nástrojů).

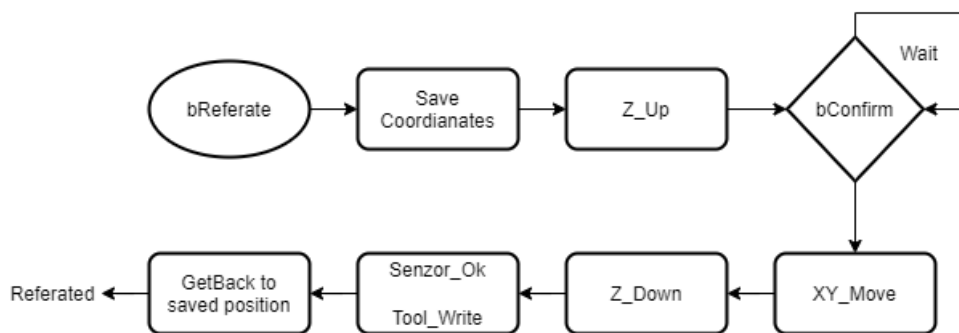
Zápis do tabulky nástrojů je prováděn funkcí „ItpWriteToolDescEx();“ Pro fungování této funkce je nutné mít sestavenou interpolační skupinu. Argumenty popsány níže:

```
fbWriteTool(bExecute:= , nDNo:= , tTimeOut:= , sNciToPlc:= , sToolDesc:= ,  
bBusy=> , bErr=> , nErrId=> );
```

- **bExecute** – aktivace na náběžnou hranu.
- **nDNo** – číslo zvoleného nástroje.
- **sNciToPlc** – struktura pro komunikaci s interpolační skupinou.
- **sToolDesc** – speciální struktura parametrů nástroje. Například parametr „2“ je délka nástroje a parametr „3“ je jeho poloměr.

Obdobná funkce „ItpReadToolDesc();“ je použita pro výčet parametrů nástroje. Pro výčet parametrů nemusí být sestavena interpolační skupina.

Při výběru nástroje je nutné počítat s tím, že jeho číslo je rovno průměru nástroje. Délková korekce je určena senzorem automaticky. Stavový automat k určení délkové korekce nástroje je na obrázku 6.7.



Obr. 6.7: Stavový automat určení délkové korekce nástroje

Automat je spuštěn tlačítkem „Change tool“, poté se zapamatují aktuální souřadnice a vřeteno vyjede nahoru. Čeká se potvrzení výměny „Refer“. Následně vřeteno najede nad senzor a pomalu klesá, dokud nerozepne a nezměří jeho délku. Ta je zapsána do příslušné struktury a vřeteno opět vyjede nahoru. Posledním krokem je posuv zpátky na startovní pozici a nástroj je zreferován.

6.7 Programování informačního kanálu

Informační kanál, tedy levá střední sekce Obr. 6.1: Návrh vizualizace, je složen ze čtyř částí. První část „Machine“ zobrazuje 3D-model přestavěné frézky. Druhá část „Chart“ slouží k vykreslení pohybu nástroje. Další, třetí část „Info“ obsahuje prostředky vypisující různé parametry terminálů. Poslední, čtvrtá část „EventLog“ zobrazuje tabulku událostí vztahující se ke stroji.

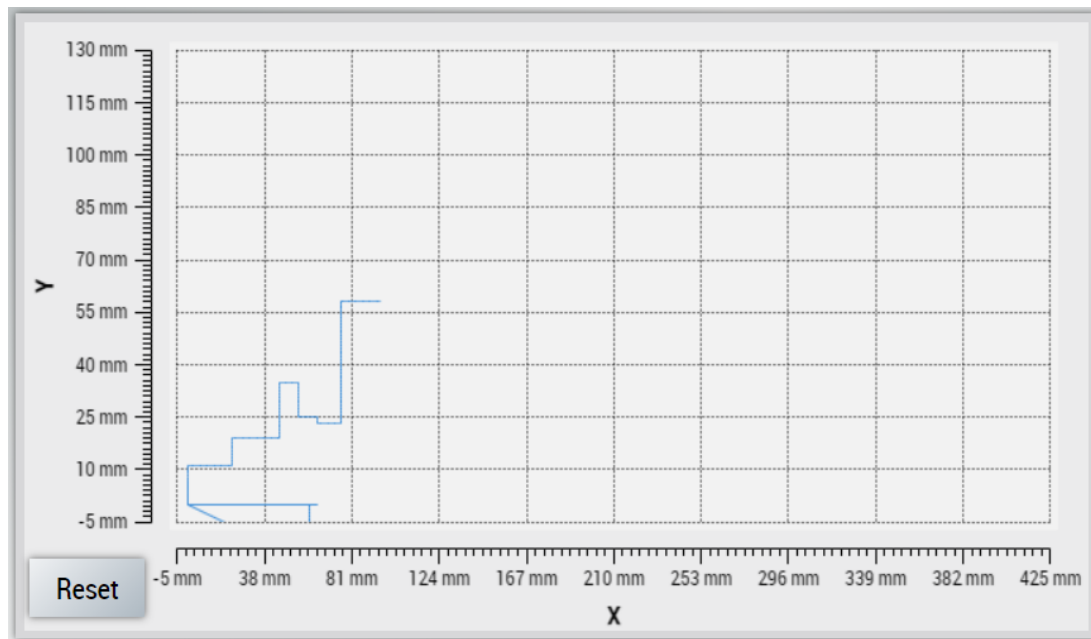
6.7.1 Vykreslování pohybu

Vykreslování pohybu je realizováno v programu „PRG_Graf“. HMI umožňuje vykreslení hodnoty v čase nástrojem „Line chart“. Pro zobrazení „XY“ grafu je potřeba tento prvek nastavit a předávat mu jako parametr pole hodnot osahující strukturu souřadnic [x,y]. Programově je daný prvek tohoto pole plněn vždy aktuální pozicí dané osy. Předěšlý prvek je pomocí cyklu „FOR“ posunut v poli vždy o jedno místo zpět. Počet iterací cyklu je rovna velikosti pole.

Hodnota aktuální pozice osy je převedena na celé číslo pomocí příkazu „LREAL_TO_INT();“ a do pole prvků je zapisována až poté, co je detekován rozdíl

mezi minulou a aktuální souřadnicí větší nebo roven jedné. Tato metoda umožňuje snížit počet prvků pole a vůči použité velikosti grafu téměř neovlivní přesnost.

Graf je také doplněn resetovacím tlačítkem, které pole prvků vyplní aktuální hodnotou os a tím nahradí aktuální křivku grafu jedním aktuálním bodem, tedy aktuální pozicí. Příklad vykreslení je na obrázku 6.8.



Obr. 6.8: Vykreslení pohybu

6.7.2 Informace o motorech a terminálu

Program „PRG_Inf“ zprostředkovává vyčtení parametrů motorů a terminálu z daného „EL7041“ terminálu. V HMI je možné zvolit parametr prvkem „ComboBox“ a parametr se následně zobrazí pro všechny jednotlivé osy.

Pro vyčítání parametrů je vytvořen funkční blok „FB_InfoRead“, který používá funkční blok „FB_EcCoESdoRead()“, avšak eliminuje počet argumentů vyřazením oněch neměnných. Argumenty a jejich funkce popsány níže:

```
fbInfoRead(sNetId:= , nSlaveAddr:= , nSubIndex:= , nIndex:= , pDstBuf:= ,
cbBufLen:= , bExecute:= , tTimeout:= , bBusy=> , bError=> , nErrId=> ,
cbRead=> );
```

- **sNetId** – AmsNetId runtimeu předané jako „string“.
- **nSlaveAddr** – Adresa terminálu, ze kterého je parametr vyčítán. Cesta: „Term 7 -> EtherCAT -> EtherCAT Addr“ a hodnota „1005“.
- **nSubIndex** – pod index vyčítaného parametru, zapsán šestnáctkově.

Například: „16#04“.

- **nIndex** – index vyčítaného parametru, zapsán šestnáctkově.
- **pDstBuf** – předání adresy proměnné pro uložení vyčtené hodnoty.
Použití funkce „ADR();“
- **cbBufLen** – předání velikosti proměnné pro uložení vyčtené hodnoty.
Použití funkce „SIZEOF();“
- **bExecute** – aktivace na náběžnou hranu.

Seznam vyčítaných parametrů a jejich funkcí je uveden v tabulce 6.1.

Tab. 6.1: Vyčítané informace o terminálu EL7041

Index	SubIndex	Název	Jednotka
0x9010	01	Status word	-
	02	Motor coil voltage A	mV
	03	Motor coil voltage B	mV
	04	Motor coil current A	mA
	05	Motor coil current B	mA
	06	Duty cycle A	%
	07	Duty cycle B	%
0x9020	04	Actual position lag	-
0xF900	02	Internal terminal temp.	°C
	04	Control voltage	mV
	05	Motor supply voltage	mV

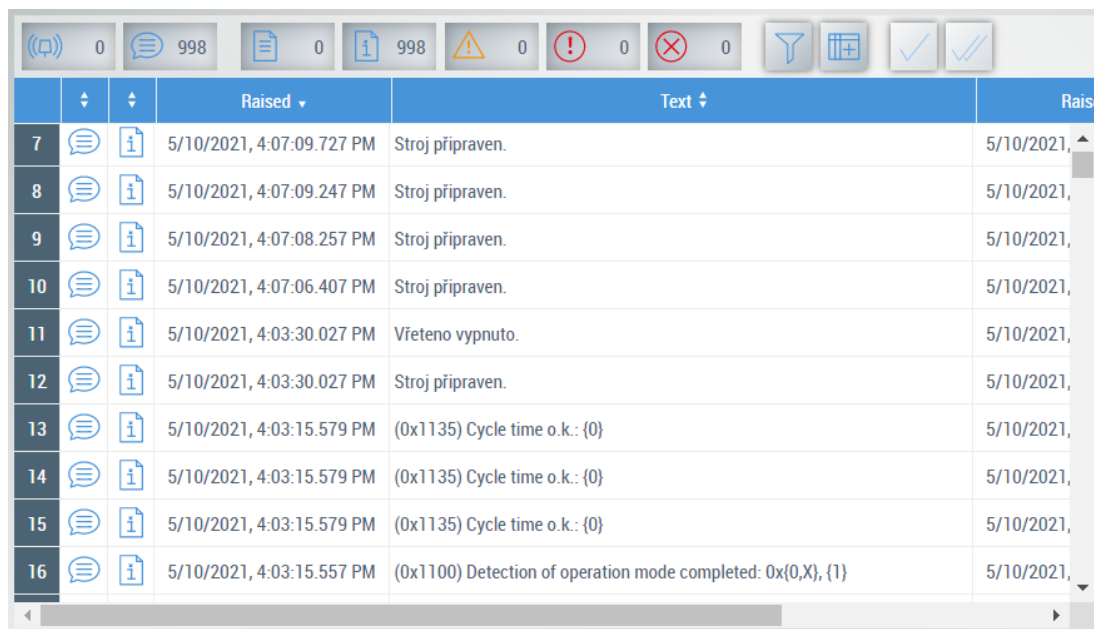
6.7.3 Eventlogger

„Event Grid“ je nástroj HMI pro zobrazení informací z „TwinCAT Logged Events“ (EventLogger). Zobrazený nástroj je na obrázku 6.9. Možné je zobrazit události TwinCAT nebo si vytvořit vlastní seznam událostí v závislosti na PLC, popřípadě obojí.

Vytvoření třídy pro události se provede: „SYSTÉM -> Type System -> Event Classes“. Zde po kliknutí pravého tlačítka a „New“ se vytvoří požadovaná třída. Ve třídě je možné vytvořit události různých typů, jako např. „Info“ nebo „Warning“.

V programu jsou následně vytvořeny instance alarmů či zpráv skrze „FB_TcAlarm.CreateEx();“ a „FB_TcMessage.CreateEx();“. V prvním argument těchto funkcí je zapotřebí odkazovat se na danou událost. Toto je provedeno přes „Tc_Events.<NázevVytvořenéTřídy>.<NázevVytvořené události>“.

Alarmy jsou volány „FB_TcAlarm.Raise(0);“. Zprávy jsou volány podobným způsobem „FB_TcMessage.Send(0);“. Po vyvolání těchto akcí se daný alarm nebo daná zpráva zobrazí v „Eventloggeru“. Vytvořeny jsou 4 alarmy a 20 zpráv, odkazující se na události při ovládání stroje.



			Raised	Text	Raised
7		i	5/10/2021, 4:07:09.727 PM	Stroj připraven.	5/10/2021,
8		i	5/10/2021, 4:07:09.247 PM	Stroj připraven.	5/10/2021,
9		i	5/10/2021, 4:07:08.257 PM	Stroj připraven.	5/10/2021,
10		i	5/10/2021, 4:07:06.407 PM	Stroj připraven.	5/10/2021,
11		i	5/10/2021, 4:03:30.027 PM	Vřeteno vypnuto.	5/10/2021,
12		i	5/10/2021, 4:03:30.027 PM	Stroj připraven.	5/10/2021,
13		i	5/10/2021, 4:03:15.579 PM	(0x1135) Cycle time o.k.: {0}	5/10/2021,
14		i	5/10/2021, 4:03:15.579 PM	(0x1135) Cycle time o.k.: {0}	5/10/2021,
15		i	5/10/2021, 4:03:15.579 PM	(0x1135) Cycle time o.k.: {0}	5/10/2021,
16		i	5/10/2021, 4:03:15.557 PM	(0x1100) Detection of operation mode completed: 0x{0,X}, {1}	5/10/2021,

Obr. 6.9: HMI zobrazení událostí

6.8 Ostatní vytvořené funkční vybavení

V celém projektu je použita celá řada dalších algoritmů a programových vybavení, z nichž některé jsou popsány níže.

6.8.1 Vypnutí a restart IPC

Korektní vypnutí či restartování průmyslového počítače z vizualizace je důležitou součástí projektu. Zde je tato funkce realizována pomocí funkčního bloku „NT_StartProcess();“ a použití příkazového řádku společně s „.bat“ soubory. Popis důležitých argumentů funkčního bloku:

```
fbStartProcess(NETID:= , PATHSTR:= , DIRNAME:= , COMNDLINE:= , START:= ,
TMOUT:= , BUSY=> , ERR=> , ERRID=> );
```

- **NETID** – „AmsNetId“ počítač, kde běží „runtime“.

- **PATHSTR** – cesta k souboru, který chceme spustit, včetně jeho názvu. V našem případě se jedná o „bat“ soubory se složce „C:\Bat\“.
- **DIRNAME** – cesta do adresáře na zástupce „cmd“, např. „C:\Bat\“.
- **CMDLINE** – spuštění příkazového řádku „cmd.exe“.
- **START** – aktivace funkčního bloku náběžnou hranou.

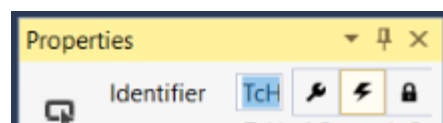
Obsah zdrojových souborů je složen z několika příkazů, z nichž nejdůležitější je příkaz „SHUTDOWN“ a jeho parametry. První parametry určuje, zda-li se jedná o vypnutí počítače „-s“ nebo jeho restartování „-r“.

6.8.2 Vybavení vizualizace

Celá vizualizace je rozdělena na kontenty „<název>.content“. Tyto kontenty jsou zobrazeny prvkem „Region“. Použití tohoto uspořádání zpřehledňuje vizualizaci a dává prostor k vizualizacím různých velikostí.

Ve vizualizaci je použita celá řada nástrojů jako jsou tlačítka „Button“, textové kontejnery „TextBlock“ a „TextBox“, nástroje možnosti výběru „ComboBox“ a další. U všech těchto prvků lze nastavit události „Events“, obrázek. 6.10 ikona blesku.

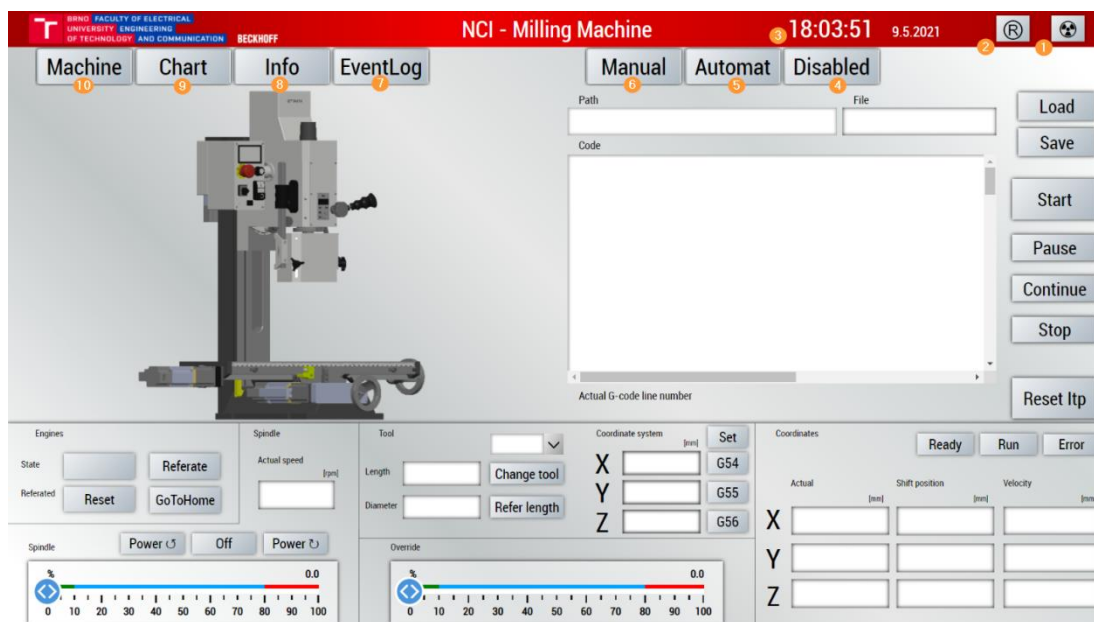
Zde je možno nastavit události na kliknutí, dvojkliknutí, na změnu stavu a další. Tyto události jsou v projektu často využity.



Obr 6.10: Eventy HMI

7 POPIS OVLÁDÁNÍ

Tato kapitola popisuje ovládání stroje. Popsáno je zde pohotovostní ovládání z rozvaděče a hlavní částí je zde popis ovládání řízení z vizualizace a popis ovládání doprovodných funkcí k řízení.



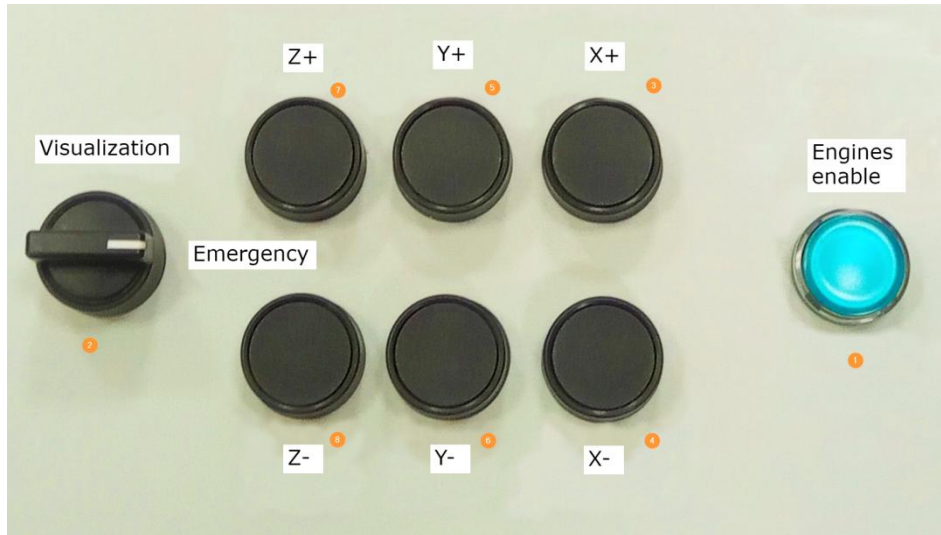
Obr. 7.1: Náhled na celkovou vizualizaci

Náhled na celkovou vytvořenou vizualizaci, jež zaštiťuje potřebné řízení, je na obrázku 7.1. Popis obrázku značí:

- **1:** Tlačítko pro vypnutí IPC.
- **2:** Tlačítko pro restart IPC.
- **3:** Zobrazení aktuálního času a aktuálního data.
- **4:** Výběr ovládání – motory vypnuty. V případě výběru toho řízení, je možné použít frézku ručně a obrábět pomocí kliček. Ve vizualizaci se bude stále zobrazovat aktuální pozice, avšak motory generují určité zpětné napětí.
- **5:** Výběr ovládání – automatické řízení, pomocí g-kódu.
- **6:** Výběr ovládání – manuální řízení.
- **7:** Výběr zobrazovaných informací – události ovládání -> „EventLogger“.
- **8:** Výběr zobrazovaných informací – záložka informací o stroji.
- **9:** Výběr zobrazovaných informací – vykreslení pohybu stroje.
- **10:** Výběr zobrazovaných informací – záložka informací o stroji.

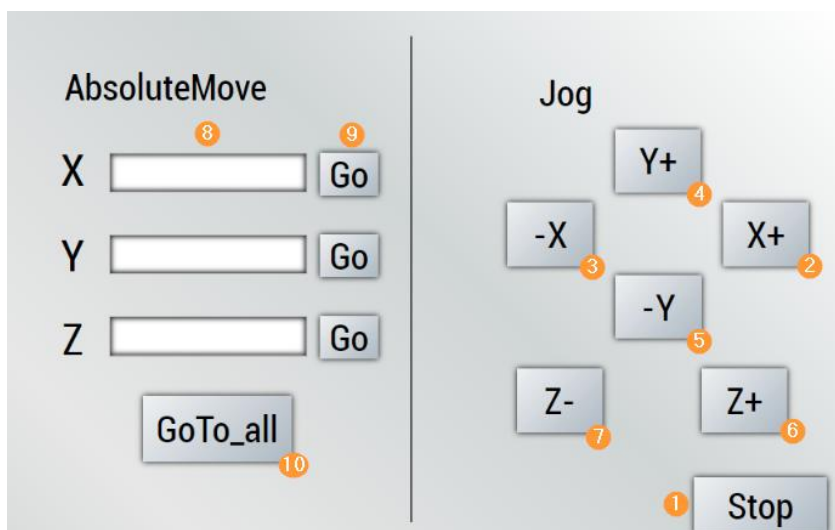
7.1 Popis pohotovostního ovládání

Pohotovostním ovládáním je myšleno ovládání tlačítka z rozvaděče. Ovládání je zobrazeno na obrázku 7.2. Jednotlivé body na obrázku mají následující funkci:



Obr. 7.2: Pohotovostní ovládání

- **1:** Tlačítko, s funkcí led, pro vypnutí a zapnutí motorů.
- **2:** Přepínač výběru řízení:
 - horizontální – zvoleno pohotovostní řízení.
 - vertikální – zvoleno řízení z vizualizace.
- **3:** Posuv X+
- **4:** Posuv X-
- **5:** Posuv Y+
- **6:** Posuv Y-
- **7:** Posuv Z+
- **8:** Posuv Z-



Obr. 7.3: Ovládání manuálního řízení

7.2 Popis vizualizace – manuální ovládání

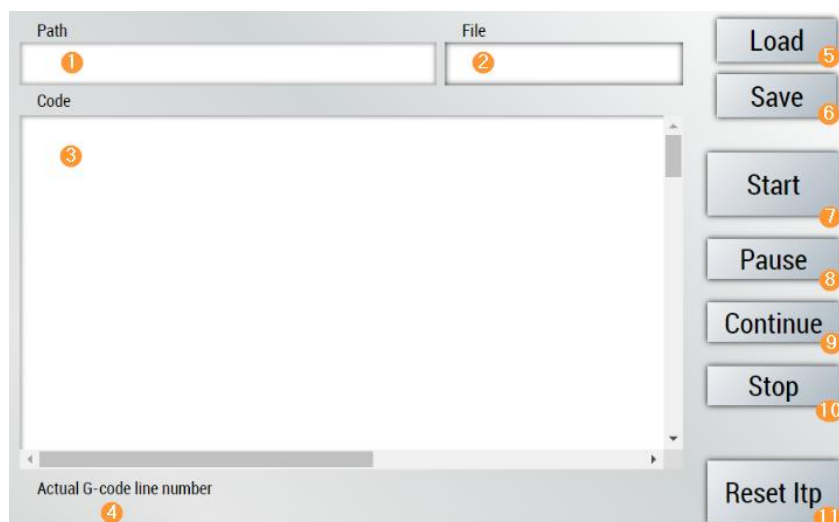
Manuální ovládání z HMI je na obrázku 7.3. Označení na obrázku mají tyto funkce:

- **1:** Stop – zastavení pohybu motorů
- **2:** Posuv X+
- **3:** Posuv X-
- **4:** Posuv Y+
- **5:** Posuv Y-
- **6:** Posuv Z+
- **7:** Posuv Z-
- **8:** Zápis cílových souřadnic [X,Y,Z], na které má být proveden posuv.
- **9:** Potvrzení (spuštění) posuvů jednotlivých os.
- **10:** Potvrzení posuvu všech os na cílové souřadnice najednou.

V rámci manuální řízení je možné ovládat posuvy také pomocí ručního ovladače. U tohoto ovladače je nutné stisknout tlačítko na boku, zvolit osu a velikost posuvu. Poté je velikost posuvu načítána inkrementací, popř. dekrementací, ručního kola.

7.3 Popis vizualizace – automatické ovládání

Ovládání automatického řízení, tedy řízení pomocí g-kódu, je vyobrazeno na obrázku 7.4. Jednotlivé funkce ovládání jsou popsány následně.



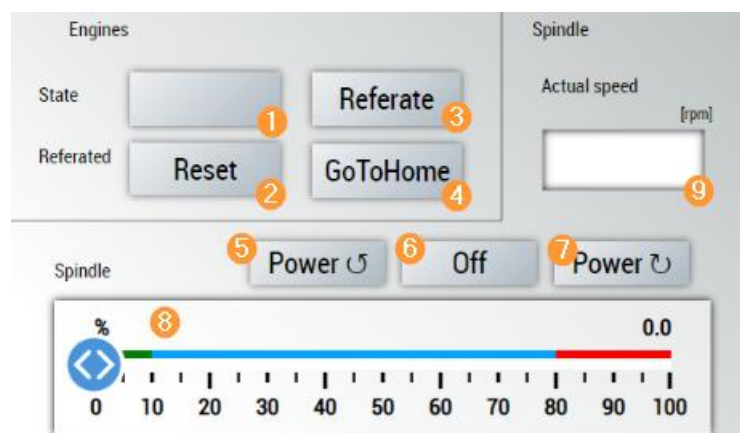
Obr. 7.4: Ovládání automatického řízení

- **1:** **Path** – cesta ke zdrojovému souboru. Nutné nahrávat „.nc“ soubory do jedné složky, popřípadě manuálně měnit tuto cestu.
- **2:** **File** – název vybraného souboru, generován automaticky.
- **3:** **Code** – vypsání zdrojového kódu načteného souboru.

- **4:** **Act. line number** – číslo aktuálního prováděného bloku kódu.
- **5:** **Load** – načtení zdrojového souboru. Po kliknutí se otevře dialogové okno, ve kterém je možné vybrat zdrojový soubor.
- **6:** **Save** – uložení aktuálního obsahu okna „Code“. Po kliknutí se také otevře dialogové okno, jenž dá na výběr kam a jak soubor uložit.
- **7:** **Start** – spuštění automatického obrábění.
- **8:** **Pause** – pozastavení automatického obrábění.
- **9:** **Continue** – pokračování po pozastavení.
- **10:** **Stop** – zastavení, obrábění nedokončeno.
- **11:** **Reset ltp** – reset interpolační skupiny. Použití v případě chyby v průběhu automatického chodu.

7.4 Popis ovládání doprovodných prvků

Popis ovládání doprovodných prvků zaštiťuje celý spodní pás ovládání z obr. 7.1. Pro popis je tento pás rozdělen do tří částí, které jsou i s jejich funkcemi popsány na následujících obrázcích 7.5, 7.6 a 7.7.

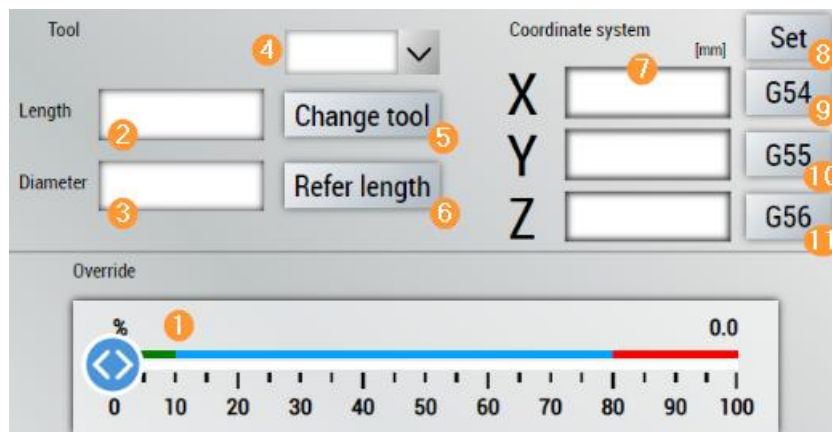


Obr. 7.5: Ovládání motorů a vřetene

- **Engines** – ovládání motorů.
 - **1:** Zobrazení stavu motorů -> „Enabled/Disabled“.
 - **2:** **Reset** – reset motorů.
 - **3:** **Referate** – zreferování os na čidla.
 - **4:** **GoToHome** – posuv na domovskou pozici [0,0,0].
- **Spindle** – ovládání vřetena.
 - **5:** **Power pro** – zapnutí vřetene ve směru hod. ručiček.
 - **6:** **Off** – vypnutí vřetene

- **7:** **Power con** – zapnutí vřetene proti směru hod. ručiček.
- **8:** Posuvník nastavení otáček vřetene.
- **9:** Aktuální hodnota otáček vřetene.

Ovládání vřetene je naprogramováno, ale ještě není fyzicky připojeno.



Obr. 7.6: Override, funkce nástroje a souřadnicový systém

- **Override** – nastavení výkonu posuvu.
 - **1:** Volba rychlosti posuvu – max. 10mm/s.
- **Tool** – nastavení korekce nástroje.
 - **2:** Aktuální délková korekce použitého nástroje.
 - **3:** Aktuální průměr použitého nástroje
 - **4:** Výběr nástroje. Číslo nástroje je rovno jeho průměru.
 - **5:** **Change tool** – Zahájení výměny nástroje. Vřeteno povyjede nahoru v ose Z.
 - **6:** **Refer length** – Potvrzení výměny nástroje a následuje jeho automatické referování a odečtení délkové korekce.
- **Coordinates shift** – posun či výběr různých souřadnicových systémů. Použitelné pro určení počátku obrobku.
 - **7:** Aktuální hodnota posuvu souřadného systému.
 - **8:** **Set** – nastavení aktuálních souřadnic jako počátek souřadnicového systému.
 - **9:** **G54** – zvolení souřadnicového systému G54, tento systém nemůže být měněn a je absolutně nastaven na [0,0,0].
 - **10:** **G55** – zvolení souřadnicového systému G55, může být měněn.
 - **11:** **G56** – zvolení souřadnicového systému G56, může být měněn.

Coordinates

Ready 4

Run 5

Error 6

	Actual [mm]	Shift position [mm]	Velocity [mm/s]
X	1	2	3
Y			
Z			

Obr. 7.7: Aktuální souřadnice, rychlosti a stavy os

- **Coordinates** – hodnoty os.
 - **1:** Aktuální hodnota souřadnic.
 - **2:** Aktuální hodnota souřadnic vůči původnímu souřadnému systému.
 - **3:** Aktuální rychlost jednotlivých os.
- **States** – stavy stroje.
 - **4:** **Ready** – zelená barva, značí připravený stroj na řízení.
 - **5:** **Run** – žlutá barva, značí stroj právě v automatickém chodu.
 - **6:** **Error** – červená barva, značí stroj v chybě.

8 UVEDENÍ DO PROVOZU

Tato část popisuje uvedení jak hardwarové, tak i softwarové části do provozu. Je zde také popsáno testování zařízení a oprava chyb s tím spojená.

První spuštění stroje proběhlo korektně bez nutnosti zásahu do elektroinstalace. Rozvaděč byl ze zadní strany také doplněn dvěma zásuvkami na din, do kterých bylo připojeno napájení vřetene a dotykové obrazovky tak, aby bylo možné ovládat napájení všech komponent frézky jedním hlavním vypínačem.

Následně bylo otestováno spuštění vizualizace po stratu počítače, které proběhlo správně s požadovaným zpožděním oproti „runtimu“.

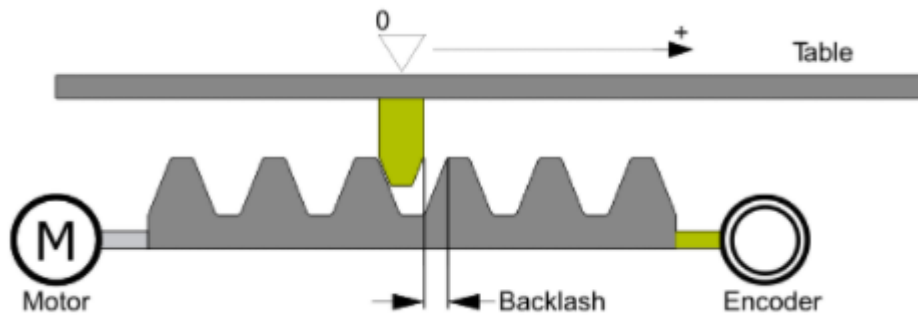
Spuštění či vypnutí motorů v různých druzích ovládání řízení bylo také úspěšně otestováno. Po tomto kroku přišlo testování samotných jednoduchých pohybů, které v jednom případě bylo v pořádku, v druhém případě byl pohyb invertován a ve třetím se rozházel kladný směr otáčení motoru vůči enkodéru a enkodér posílal do zpětné vazby převrácené data, jejichž důsledkem se snažilo řízení vyregulovat chybu posuvu, které byla ovšem tímto počinem zvyšována. Tyto chyb byly vyřešeny invertováním polarit motorů nebo enkodérů, či obojím. Následující tabulka 8.1 vypisuje výsledné polarity motorů a enkodérů použité v programu při daném elektrickém zapojení.

Tab. 8.1: Polarity motorů a enkodérů v programu

Axis	Engine polarity	IRC polarity
X	TRUE	FALSE
Y	FALSE	TRUE
Z	TRUE	FALSE

Dalším krokem po opravě směru os bylo znovu testování pohybů a to jak kontinuální pohyb, tak i absolutní pohyb na konkrétní souřadnice. V rámci tohoto testování byl zařazen také ruční ovladač, jež fungoval dobře, a do něhož byla následně do stop tlačítka doplněna funkce resetu motorů, která dokázala zastavit posuv stroje v jakémkoliv režimu.

V průběhu zkoušení motorů byla zjištěna jistá vůle v převodu v ose Y. Tato vůle je nejspíše způsobena přenesením pohybu z trapézového šroubu na křížový stul, kde se vyskytuje jistá benevolence při změně směru, viz obr. 8.1.



Obr. 8.1: Vůle v převodu osa Y [37]

Tento nežádoucí jev byl prakticky téměř anulován použitím funkčního „MC_BacklashCompensation“, který je volán cyklicky a popsán následně:

```
fb_BacklashComp( Axis:= , Enable:= , Backlash:= ,  
CompensationInPositiveDirection:= , Ramp:= , DisableMode:= , Options:= ,  
Enabled=> , Busy=> , Error=> , ErrorID=> , CurrentBacklash=> , Limiting=> );
```

- **Axis** – reference na používanou osu.
- **Enable** – aktivní na stav „TRUE“
- **Backlash** – velikost vůle, v tomto případě 0,3 mm.
- **CompensationInPositiveDirection** – „TRUE/FALSE“ v závislosti na tom o jakou vůli v převodu se jedná, viz infosys.
- **Ramp** – Nastavená rychlost dorovnání pozice.

Po odladění základních pohybů byly stanoveny limity jednotlivých os. Tyto maximální a minimální pozice se nachází v Tab. 5.3: Nastavené parametry „Enc“.

Načítání a ukládání „nc“ kódu bylo funkční. Testování automatického obrábění bylo provedeno na základním kódu a kódu vygenerovaného 2D simulátorem. Spuštění a provedení těchto kódů bylo úspěšně otestováno. Nesrovnalost nastala při generaci kódu z CAM procesoru, jež generoval určité netisknutelné znaky a g-funkce, které nejsou použitým interpolátorem podporovány.

ZÁVĚR

Tato diplomová práce se zabývala přestavěním manuální vrtačkofrézky BF20L na CNC frézku i s realizací elektroinstalace a řízení pomocí PLC. Součástí této práce byl také teoretický rozbor CNC stroje, jeho částí a také částí spojených s přestavbou a řízením frézky. Dále zde byl uveden výchozí stav frézky.

Další sekce se zabývala zvolením vhodných komponent pro přestavbu frézky a zejména její řízení. K přestavbě byl využit speciální „kit“ pomocí kterého jsou všechny osy poháněny krokovými motory doplněné o enkodéry. Frézka byla doplněna limitními snímači. Pro jejich uchycení byly vytvořeny modely v programu Solidworks, společně s 3D modelem celé frézky. Pro řízení byl použit průmyslový počítač C6015-0010 se softwarovým PLC společně s I/O terminály a terminály pro řízení krokových motorů EL7041 od společnosti Beckhoff.

Tato hardwarová přestavba byla implementována na frézku společně s elektronikou řízení, která byla zabudována do průmyslových rozvaděčů. Vytvořeno bylo i elektrotechnické schéma zapojení v programu ProfiCAD. Celá frézka společně s rozvaděči byla upevněna ke stolu, tudíž společně budí dojem jednoho stroje.

Programové řízení bylo vytvořeno v softwaru TwinCAT3 a jeho funkcích. Naprogramováno bylo pohotovostní řízení tlačítka z rozvaděče, manuální řízení za pomoci vizualizace, jehož součástí je i ruční ovladač, a dále také automatické řízení, které umožní obrábění pomocí předem definovaného programu. Pro toto řízení byla vytvořena vizualizace. Samotné původní ovládání frézky je stále použitelné při deaktivaci motorů. Na vytvoření programu byl použit standardní PLC jazyk strukturovaný text s lehkou nadstavbou objektového programování. Vytvořená vizualizace řízení byla doplněna funkcemi, vytvořenými jazykem JavaScript.

Z hlediska možného vylepšení je v plánu napojení průmyslového počítače do školní sítě, odkud bude možné na stroj nahrávat příslušné programy. Dále pak finální připojení vřetene, jehož řídicí deska bude muset být doplněna o dps s ovládáním otáček a přepínáním mezi ovládáním z frézky a PLC. Jako poslední bod bych doporučil detailnější ošetření chybových stavů a vyvarování se před nimi.

V praxi tento stroj najde své uplatnění při použití na menší jednokusovou výrobu, tedy v malých hobby provozech.

SEZNAM LITERATURY

- [1] ŠTULPA, Miloslav. *CNC: programování obráběcích strojů*. Praha: Grada, 2015. ISBN 978-80-247-5269-3.
- [2] *Obscný úvod do problematiky CNC programování* [online]. 75 [cit. 2020-10-15]. Dostupné z: https://www.sosbites.cz/images/stories/Pro-studenty/studijni-materialy/VUKOV_TEXT_-_1.ST.pdf
- [3] Mgr. Václav Němec a Ing. Stanislav Dlouhý. *Programování CNC obráběcích strojů* [online]. 2013, 67 [cit. 2020-10-15]. Dostupné z: https://www.gvi.cz/Aton/FileRepository/aton_file_repository_HtmlEditorRepositoryDoc/Root/Projekty/Cnc.pdf
- [4] *Vitralab: Příručka CNC programování* [online]. 31 [cit. 2020-10-16]. Dostupné z: http://www.sjf.tuke.sk/vitralab/upload/CNC%20prirucka_CZ.pdf
- [5] Ing. Petter Keller, PhD. *Programování a řízení CNC strojů* [online]. 2005, 51 [cit. 2020-10-16]. Dostupné z: http://www.kvs.tul.cz/download/cnc_cadcam/pnc_2.pdf
- [6] CAD/CAM. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2020 [cit. 2020-10-17]. Dostupné z: <https://en.wikipedia.org/wiki/CAD/CAM>
- [7] MAREK, Jiří. *Konstrukce CNC obráběcích strojů III*. Praha: MM publishing, 2014. MM speciál. ISBN 978-80-260-6780-1.
- [8] *Siemens* [online]. 1996-2021 [cit. 2020-10-24]. Dostupné z: <https://new.siemens.com/global/en/products/automation/systems/cnc-sinumerik/automation-systems/sinumerik-840.html>
- [9] *Newfangled Solutions: Mach4* [online]. Kiseki Studio, 2021 [cit. 2020-10-24]. Dostupné z: <https://www.machsupport.com/software/mach4/>
- [10] *Beckhoff: New Automation Technology* [online]. Beckhoff Automation, 2020 [cit. 2020-10-24]. Dostupné z: <https://www.beckhoff.com/>
- [11] *Návod k obsluze, Vrtáčko - frézka BF 20 L Vario*. 3.1.0. Germany: Optimum Maschinen Germany, 2012 [cit. 2020-11-14].
- [12] Radovan Páv. *Značkové nářadí* [online]. 2011 [cit. 2020-11-14]. Dostupné z: <https://www.znackove-naradi.eu/stolni-frezka-optimill-bf-20-vario-107028.html#technical>
- [13] *Obzor: Eshop* [online]. Zlín: Surface, 2015 [cit. 2020-12-16]. Dostupné z: <https://www.obzor.cz/eshop/vackovy-spinac-vsn-10-a>
- [14] *Conrad: MeanWell - datasheet* [online]. 2015 [cit. 2020-12-16]. Dostupné z: <https://asset.conrad.com/media10/add/160267/c1/-/en/001439435DS01/list-technickych-udaju-1439435-mean-well-ndr-480-24-sitovy-zdroj-na-din-listu-480-w-1-x.pdf>
- [15] *ElektroSMS: ŘADOVÁ SVORKA POJISTKOVÁ RSP 4 LED/24-48V* [online]. České budějovice, 2020 [cit. 2020-12-16]. Dostupné z:

<https://shop.elektrosms.cz/cs/radova-svornice-pojistkova-rsp-4-led-24-48v-zadny-vyrobce-skl000012543>

- [16] *MachineDesign: What's the Difference Between Servo and Stepper Motors?* [online]. Endeavor Business Media, 2021 [cit. 2020-12-17]. Dostupné z: <https://www.machinedesign.com/mechanical-motion-systems/article/21836868/whats-the-difference-between-servo-and-stepper-motors>
- [17] *Sew-Eurodrive: Servomotory* [online]. 2021 [cit. 2020-12-17]. Dostupné z: https://www.sew-eurodrive.cz/vyrobky/motory/servomotory/servomotoren_3.html
- [18] *E-konstruktor* [online]. 2013 [cit. 2020-12-18]. Dostupné z: <https://e-konstrukter.cz/novinka/zakladni-principy-krokovych-motoru>
- [19] Doc. Ing. Josef Formánek, Ph.D. *Výukové texty* [online]. 2014, (09), 13 [cit. 2020-12-18]. Dostupné z: <http://home.zcu.cz/~formanek/mmvyuka-amzt/Data/ivk-arvt-soubory/09-F.pdf>
- [20] *Parker Hannifin: SERIES VS PARALLEL WIRING IN MOTORS* [online]. 2021 [cit. 2020-12-19]. Dostupné z: <https://community.parker.com/technologies/electromechanical-group/w/electromechanical-knowledge-base/200/general---series-vs-parallel-wiring-in-motors>
- [21] *Microcon: Zapojení vinutí jednotlivých tipů krokových motorů* [online]. [cit. 2020-12-19]. Dostupné z: <http://www.microcon.cz/zapojenivinuti2012web/vinutihlstranka10.asp>
- [22] *Joy-it: NEMA23-01-Datasheet* [online]. 2019 [cit. 2020-12-21]. Dostupné z: <https://joy-it.net/files/files/Produkty/NEMA23-01/NEMA23-01-Datasheet.pdf>
- [23] *CNCshop: S3806 rotační inkrementální enkodér* [online]. 2010 [cit. 2020-12-21]. Dostupné z: <http://www.cncshop.cz/s3806-rotacni-inkrementalni-enkoder>
- [24] *T-support: CNC řídicí systém je srdce i mozek CNC obráběcího stroje (I)* [online]. Praha 6, 2012 [cit. 2020-12-22]. Dostupné z: <https://www.t-support.cz/kat/cnc-ridici-system-je-srdce-i-mozek-cnc-obrabeciho-stroje-1>
- [25] KREJČÍ, Jan. *Retrofit/digitální řízení CNC manuální vrtačkofrézky BF20L* [online]. Brno, 2019 [cit. 2020-12-22]. Dostupné z: <https://www.vutbr.cz/studenti/zavprace/detail/119246>. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky. Vedoucí práce Zdeněk Bradáč.
- [26] Programmable logic controller. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2020 [cit. 2021-12-23]. Dostupné z: https://en.wikipedia.org/wiki/Programmable_logic_controller
- [27] *CNCshop: ISM1468 Ruční ovladač* [online]. 2010 [cit. 2020-12-24]. Dostupné z: <http://www.cncshop.cz/ism1468-rucni-ovladac>
- [28] *Profitek: SENZOR DÉLKY UPNUTÉHO NÁSTROJE TS32* [online]. 2021 [cit. 2020-12-24]. Dostupné z: <https://eshop.profitex.cz/pracovni-jednotky/senzor->

odmerovani-delky-nastroje-ts32/

- [29] *Ifm: IES202* [online]. 2021 [cit. 2021-3-4]. Dostupné z: <https://www.ifm.com/us/en/product/IES202>
- [30] *ASUS: VT168H* [online]. [cit. 2021-3-5]. Dostupné z: <https://www.asus.com/cz/Displays-Desktops/Monitors/All-series/VT168H/>
- [31] *GrabCAD* [online]. 2021 [cit. 2021-3-10]. Dostupné z: <https://grabcad.com/>
- [32] *Beckhoff: TwinCAT 3* [online]. Germany, 2019 [cit. 2021-3-28]. Dostupné z: https://www.beckhoff.com/media/downloads/information-media/beckhoff_twincat3_e.pdf
- [33] *TwinCAT 3: eXtended Automation Technology (XAT)*. *AUTOMA: časopis pro automatizační techniku* [online]. 2015, **2015**(06) [cit. 2021-3-28]. Dostupné z: https://automa.cz/cz/casopis-clanky/twincat-3-extended-automation-technology-xat-2015_06_53801_6979/
- [34] *TE2000 TC3 HMI Engineering: Manual - TwinCAT 3* [online]. 2018 [cit. 2021-4-2]. Dostupné z: https://automa.cz/cz/casopis-clanky/twincat-3-extended-automation-technology-xat-2015_06_53801_6979/
- [35] *EtherCAT: EtherCAT - the Ethernet Fieldbus* [online]. 2003 [cit. 2021-4-3]. Dostupné z: <https://www.ethercat.org/en/technology.html>
- [36] *Jak se ve skutečnosti vybírají průmyslové sběrnice*. *AUTOMA: časopis pro automatizační techniku* [online]. 2017, **2017**(01) [cit. 2021-4-3]. Dostupné z: https://www.automa.cz/cz/casopis-clanky/jak-se-ve-skutecnosti-vybiraji-prumyslove-sbernice-2017_01_0_9491/
- [37] *Beckhoff: New Automation Technology* [online]. [cit. 2021-4-12]. Dostupné z: <https://infosys.beckhoff.com/>

SEZNAM PŘÍLOH

- **TwinCAT projekt** – „NC_MillingMachine.tnzip“ – obsahující naprogramované řízení v jednom řešení (včetně PLC projektu a vizualizace).
- **3D model frézky** – retrofit vytvořený v programu Solidworks.
- **Schéma zapojení** – ProfiCAD projekt a vygenerované schéma zapojení do PDF.
- **KiCAD projekt** – návrh přepojení motorů